

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Вінницький державний технічний університет

Р. Н. Кветний

МЕТОДИ КОМП'ЮТЕРНИХ ОБЧИСЛЕНЬ

Затверджено Ученою радою Вінницького державного технічного університету як навчальний посібник для студентів спеціальності 7.091401 “Системи управління і автоматики”. Протокол № 11 від 1 липня 2001 року

Вінниця ВДТУ 2001

УДК 681.3 (07)
К 32

Рецензенти

В. В. Кухарчук, доктор технічних наук, доцент
П. А. Молчанов, доктор технічних наук, доцент
І. І. Хаймзон, доктор технічних наук, доцент

Рекомендовано до видання Ученою радою Вінницького державного технічного університету Міністерства освіти і науки України.

Кветний Р. Н.

К 32 **Методи комп'ютерних обчислень.**

Навчальний посібник. – Вінниця: ВДТУ, 2001. – 148 с.

В книзі розглянуто головні застосування чисельних методів, що зустрічаються в типових інженерних та наукових задачах в галузі комп'ютеризованих систем управління. Описані методи та алгоритми для розв'язання нелінійних рівнянь та систем, диференціальних рівнянь (звичайних та задач математичної фізики), методи обробки даних (апроксимація функцій та даних, інтерполяція, чисельне інтегрування та диференціювання, статистичні методи і т. д.), методи оптимізації. В посібнику наведено багато прикладів з детальними алгоритмами та способами оцінки похибок та збіжності. Книга повинна привернути увагу студентів та аспірантів різних спеціальностей, інженерів та вчених.

Methods of computerized calculations. Roman Kvyetnyy.

This book covers the main field of numerical methods. This consideration is aimed at the use of such methods for the solution of typical engineering and scientific tasks in computerized control systems. The methods and algorithms for solution of non-linear equations and systems, differential equations (usual and mathematical physics tasks), methods of data processing (approach to functions and data, interpolation, numerical integration and differentiation, statistical methods etc.), methods of optimization are described in this book. There are many examples with detailed algorithms and rules for estimation of errors and convergence. This book is expected to be of interest to students and postgraduate students of different specialties, to engineers and scientists.

УДК 681.3 (07)
© Р. Кветний, 2001

Зміст

Передмова	6
1. Загальні принципи застосування обчислювальних методів	7
1.1. Обчислювальний експеримент	7
1.2. Види обчислювальних задач	7
1.3. Властивості обчислювальних алгоритмів	9
2. Задачі лінійної алгебри	12
2.1. Розв'язання систем лінійних рівнянь	12
2.1.1. Прямі методи	13
2.1.1.1. Метод Гаусса	13
2.1.1.2. Метод виключення Гаусса-Жордана	14
2.1.1.3. Модифікований метод Гаусса	14
2.1.1.4. Метод прогонки	16
2.1.2. Ітераційні методи	18
2.2. Визначення власних значень матриць	20
2.2.1. Постановка проблеми	20
2.2.2. Методи обчислення власних значень	22
2.2.2.1. Прямі методи	22
2.2.2.2. Ітераційні методи	22
2.2.2.3. Методи перетворень подібності	24
2.2.3. Порівняння методів визначення власних значень	26
3. Нелінійні задачі	31
3.1. Розв'язання нелінійних рівнянь	31
3.1.1. Метод половинного ділення	33
3.1.2. Метод хибного положення (хорд)	34
3.1.3. Метод Ньютона (дотичних)	35
3.1.4. Метод січних	36
3.1.5. Метод простої ітерації	37
3.1.6. Визначення комплексних коренів	38
3.2. Розв'язання систем нелінійних рівнянь	39
4. Методи розв'язання диференціальних рівнянь	46
4.1. Методи розв'язання задачі Коші	46
4.1.1. Однокрокові методи	47
4.1.2. Багатокрокові методи	52
4.1.3. “Жорсткі” задачі	55
4.1.4. Вибір методу розв'язання задачі Коші	56
4.2. Методи розв'язання крайових задач	57
4.2.1. Метод “стрілянини”	57
4.2.2. Різницеві методи	58

5. Методи розв'язання диференціальних рівнянь в частинних похідних	62
5.1. Різницевий метод	64
5.2. Розв'язання різних типів диференціальних рівнянь в частинних похідних	68
5.2.1. Еліптичні рівняння	68
5.2.2. Гіперболічні рівняння	70
5.2.3. Параболічні рівняння	71
5.3. Загальні рекомендації до розв'язання диференціальних рівнянь в частинних похідних	72
6. Методи обробки експериментальних даних	74
6.1. Інтерполяція	74
6.1.1. Різницеві методи	75
6.1.2. Інтерполяція за Лагранжем	81
6.1.3. Сплайн-інтерполяція	83
6.2. Апроксимація даних	85
6.3. Статистична обробка даних	88
6.4. Чисельне інтегрування	92
6.4.1. Формули Ньютона-Котеса	93
6.4.2. Формули Чебишева	95
6.4.3. Формула Гаусса	97
6.4.4. Алгоритм застосування чисельних методів	100
6.4.5. Метод Монте-Карло	101
6.5. Чисельне диференціювання	104
7. Методи оптимізації планування	107
7.1. Класична постановка задачі оптимізації	107
7.2. Класифікація задач оптимізації	108
7.3. Багатокритеріальна оптимізація	108
7.4. Гладка оптимізація	110
7.4.1. Умови Куна-Таккера	111
7.4.2. Чисельні методи гладкої оптимізації	112
7.4.3. Методи зведення загальної задачі оптимізації до задачі без обмежень	115
7.5. Опукла оптимізація	116
7.5.1. Простий субградієнтний метод опуклої оптимізації	118
7.5.2. Методи розтягу простору	119
7.6. Негладка оптимізація за методом координатного спуску (підйому)	119
7.7. Стохастична оптимізація	120
7.8. Лінійне програмування	120
7.8.1. Симплекс-метод	121
7.8.2. Транспортна задача	121

7.8.3. Цілочислове лінійне програмування	122
7.8.4. Загальна задача лінійної оптимізації	123
7.9. Теорія ігор	124
7.10. Динамічне програмування	126
7.11. Варіаційні задачі	127
7.12. Системна оптимізація	127
7.13. Застосування теорії графів до розв'язання оптимізаційних задач	129
7.14. Застосування активних експериментів при ідентифікації моделей	131
Список літератури	145

Передмова

Ця книжка має ціль викладення найбільш поширених в інженерній практиці задач обчислювальної математики. Автор намагався узагальнити досвід багаторічного викладання курсів з обчислювальних методів для студентів спеціальностей, що пов'язані з автоматикою, управлінням, інформаційно-вимірювальною технікою.

Методологічну основу для посібника створили праці відомих вчених В. М. Глушкова, О. О. Самарського, А. Н. Тіхонова, Г. І. Марчука, Б. П. Демидовича, І. А. Марона, Л. Коллатця, Т. Шупа, Дж. Форсайта, Дж. Мак-Кракена, У. Дорна та інших.

Попередником цієї книжки був виданий автором російською мовою в 1989 році в видавництві "Вища школа" навчальний посібник "Обчислювальні методи та застосування ЕОМ" (в співавторстві з В. Т. Маліковим). В дев'яності роки автор підготував разом з В. М. Дубовим декілька книжок (їх перелік є в списку літератури), де були висвітлені окремі розділи обчислювальних методів. В посібнику не подано програм для реалізації тих чи інших методів. Автор зосередився на методиках обчислень та алгоритмах (в різних випадках більш чи менш деталізованих), що робить матеріал менше залежним від часу та смаку програмістів.

Внаслідок того, що книгу орієнтовано на інженерів та дослідників в галузі технічних наук, увага, в першу чергу, приділялася доведенню всіх викладених методів обчислень до конкретних практичних алгоритмів. Тому в професійних математиків можуть виникнути питання щодо повноти математичних доведень. Але перевантаження теоремами та математичними перетвореннями, ускладнення позначень, на думку автора, призвели б значною мірою до послаблення практичної спрямованості як посібника в цілому, так і курсів лекцій для майбутніх інженерів, що на його базі викладаються. Розділ з методів оптимізації подано менш глибоко ніж попередні, але детальна класифікація та постановка задач роблять його достатньо корисним, а з другого боку, завершують цілісне сприйняття курсу.

Автор висловлює глибоку подяку своїм учням та співробітникам, що брали участь в підготовці книги до друку.

Глава 1. Загальні принципи застосування обчислювальних методів

1.1. Обчислювальний експеримент

Головним методом дослідження складних процесів та систем сьогодні став обчислювальний експеримент. Це поняття було введено академіком А. А. Самарським як метод та технологію досліджень, що ґрунтуються на побудові та аналізі комп'ютерної моделі об'єкта, який вивчається.

Перший крок – це побудова математичної моделі об'єкта за допомогою вибраного виду математичного опису: алгебраїчних, диференціальних, інтегральних рівнянь, дискретної математики, теорії множин, алгебри логіки та ін.

Другим кроком є вибір методу розв'язання цих моделей. Таким методом повинен бути метод наближених чи точних обчислень, що, в свою чергу, може бути алгоритмізований та реалізований на ЕОМ. Реалізація обчислювального експерименту на ЕОМ може бути як у вигляді готової, вже відомої програми чи пакету програм, так і нової програми, що складена вибраною мовою програмування. Після проведення обчислень настає етап обробки, аналізу та вивчення результатів, а також використання їх для корегування обчислювального експерименту та отримання висновків щодо поведінки об'єкта досліджень. Причому деякі складові цього процесу можуть бути відсутні чи змінені. Так, наприклад, за допомогою методів та програм обробки даних можуть аналізуватися результати натурного чи напівнатурного експериментів, а висновки використовуватися для корегування їх ходу та умов проведення.

В усіх випадках основу обчислювальних експериментів складає тріада: модель – метод – алгоритм (програма).

1.2. Види обчислювальних задач

Класифікувати та описати види обчислювальних задач можна за двома ознаками: вид математичного апарату та предметна область (фізична сутність).

За видом математичного апарату традиційно виділимо: задачі лінійної алгебри (матричні задачі, знаходження власних значень та векторів, системи рівнянь); нелінійні алгебраїчні задачі (нелінійні рівняння та системи рівнянь); розв'язання звичайних диференціальних рівнянь

(задача Коші та крайова задача); задачі математичної фізики (диференціальні рівняння в частинних похідних); методи обробки даних (апроксимація, інтерполяція, чисельне диференціювання та інтегрування, статистична обробка даних); методи оптимізації та ін.

Причому слід відзначити, що велика кількість обчислювальних задач зараз вже може бути розв'язана не тільки за допомогою алгоритмів наближених обчислень, що традиційно складали основу прикладної математики, а й шляхом застосування та алгоритмізації точних математичних обчислень. Це стало можливо разом з розвитком комп'ютерів та методів програмування. Прикладом таких алгоритмів можуть бути системи аналітичних обчислень, що входять до пакетів програм MAPLE, MATHEMATICA.

З іншого боку, можна класифікувати обчислювальні задачі за особливостями предметної області. Тут ми перелічимо, в першу чергу, задачі, що відносяться до автоматизації, управління, інформаційно-вимірювальної техніки, оскільки цей підручник орієнтовано саме на таких спеціалістів.

Розрахунок лінійних і нелінійних електричних кіл та систем управління шляхом розв'язання систем рівнянь, що отримані з застосуванням законів Ома та Кірхгофа.

Ідентифікація динамічних характеристик систем та перетворювачів з використанням різноманітних описів вхідних сигналів. Наприклад, метод Сімою, де передбачено низку операцій чисельного інтегрування.

Дослідження стійкості систем автоматичного керування. В залежності від критеріїв, що використовуються, це може зводитись, наприклад, до пошуку власних значень та векторів.

Аналіз якості систем автоматичного управління. Знаходження оптимальних керувальних сигналів шляхом вирішення алгебраїчного матричного рівняння Ріккати, до якого зводиться, наприклад, задача про оптимальний лінійний регулятор. Розв'язання рівняння Ріккати пов'язано з використанням низки перетворень та обчисленням декількох задач з лінійної алгебри: подібні перетворення (з матриці виду Гессенберга до виду Шура); пошук власних значень.

Дослідження нелінійних автоматичних систем, що часто зводяться до розв'язання нелінійних диференціальних рівнянь вищих порядків, застосування методів кусково-лінійної апроксимації та сплайнів.

Дослідження випадкових процесів в системах шляхом імітаційного (генерація, перетворення та обробка випадкових чисел) чи аналітичного моделювання (розв'язання різноманітних інтегро-диференціальних рівнянь).

Аналіз об'єктів контролю при розробці складних інформаційно-вимірювальних систем, що часто потребує розв'язання задач математичної

фізики, де знаходиться розподіл значень фізичних величин (швидкодії потоку рідини чи температури, хвильових процесів та ін.).

Планування експериментів при пошуку невідомих зв'язків та дослідженні пристроїв автоматики.

Розв'язання задач цифрової обробки сигналів, наприклад, за допомогою Фур'є-перетворень.

Оцінка похибок вимірювальних систем: чисельне диференціювання, ймовірнісний аналіз, інтервальні обчислення.

Оптимізація при проектуванні автоматичних систем та розв'язанні задач економічної кібернетики.

Можна навести приклади ще багатьох задач зі спеціальності, що названа вище, де застосовуються обчислювальні алгоритми.

Сьогодні неможливо уявити процес досліджень без використання комп'ютера та розробки оригінальних алгоритмів та програм для реалізації розрахункових та моделювальних процедур чи використання готових пакетів прикладних програм та програмних систем. Причому поле для створення нових проблемно-орієнтованих систем практично необмежене, а базу для них складають обчислювальні алгоритми, що застосовуються з урахуванням особливостей задачі, яка розглядається.

1.3. Властивості обчислювальних алгоритмів

Вище ми відзначили, що сьогодні множина методів, які використовуються для комп'ютерних обчислень, ширша ніж множина саме обчислювальних методів чи так званих методів наближених обчислень. Але головним предметом, що розглядається в цій книзі, є традиційна обчислювальна математика. Що стосується тих методів, які належать до інших математичних розділів, в першу чергу пов'язаних з отриманням точних аналітичних розв'язків, то тут ми тільки торкнемося їх застосувань в зв'язку з розвитком сучасних пакетів прикладних програм для розв'язання математичних задач. Таким чином, ми відокремимо тільки область алгоритмів, які найчастіше розглядають розв'язання дискретної задачі, що була задана як дискретна, чи виникла як результат дискретизації вихідної неперервної задачі. Простішим прикладом дискретизації може слугувати побудова різницевої схеми, що виникає при заміні диференціальних рівнянь кінцево-різницевиими співвідношеннями. При дискретизації неперервної задачі обов'язково виникає *похибка дискретизації*. Ця похибка є однією з головних складових в *похибці обчислювального методу*.

Іншою складовою методичної похибки є *похибка округлення*, яка виникає завдяки скінченній довжині розрядної сітки комп'ютера, що

призводить до наближеного представлення дійсних чисел. Величина цієї похибки визначається не тільки довжиною розрядної сітки ЕОМ, а також чутливістю конкретного алгоритму до похибок округлення.

Можна виділити також *похибку зрізання*, що виникає при заміні нескінченних рядів скінченними.

Похибки можна класифікувати за способом представлення як абсолютні та відносні.

Абсолютна похибка визначається як модуль різниці між точним A та наближеним a значеннями числа:

$$\Delta = |a - A|.$$

Відносна похибка визначається, як:

$$\delta = \frac{\Delta}{|a|}.$$

Без доведення (доведення дуже просте і для тренування його можна зробити самостійно) наведемо головні властивості арифметичних операцій з похибками:

– *абсолютна похибка суми k* (k – будь-яке ціле число) наближених чисел не перебільшує суми абсолютних похибок цих чисел

$$\Delta \left[\sum_{i=1}^k a_i \right] \leq \sum_{i=1}^k \Delta_i;$$

– *відносна похибка суми k* наближених чисел не перебільшує максимальної відносної похибки одного з цих чисел

$$\delta \left[\sum_{i=1}^k a_i \right] \leq \max_{1 \leq i \leq k} \delta_i;$$

– *відносна похибка добутку k* наближених чисел не перебільшує суми відносних похибок цих чисел

$$\delta \left[\prod_{i=1}^k a_i \right] \leq \sum_{i=1}^k \delta_i,$$

$$\text{де } \prod_{i=1}^k a_i = a_1 * a_2 * \dots * a_k.$$

Треба розділяти *локальні* та *глобальні* похибки. *Локальні* виникають на кожному конкретному кроці обчислювального алгоритму, а *глобальні* включають всі похибки, що виникли як на цьому кроці обчислень, так і на попередніх (ці похибки називають *похибками наслідування* чи *похибками розповсюдження*).

Важливим є поняття ітераційного алгоритму, що визначається як багатокроковий алгоритм, в якому кожний наступний крок виконується на основі попереднього. Причому можна відокремити суто ітераційні алгоритми за допомогою ітераційних процедур, де розглядаються розв'язки в окремих точках, та багатокрокові алгоритми, в яких розв'язок в попередній точці є основою для пошуку наступної. Збіжність є головною властивістю суто ітераційних алгоритмів: якщо вони незбіжні, то застосувати їх взагалі неможливо.

Наведемо декілька головних визначень щодо властивостей обчислювальних методів та алгоритмів.

Під *стійкістю* обчислювального алгоритму розуміється безперервна залежність розв'язку від вхідних даних.

Збіжність – це властивість алгоритму набувати як завгодно високої точності та прагнути до точного розв'язку вихідної задачі при збільшенні кроків обчислювального процесу.

Причому підвищення точності досягається зміною внутрішніх параметрів алгоритму (наприклад, максимально допустимою різницею між попереднім та наступним наближенням).

Коректність обчислювального методу – це властивість безперечного існування розв'язку задачі та забезпечення стійкості обчислювального алгоритму, що реалізує цей метод.

Контрольні запитання та завдання

1. Що таке обчислювальний експеримент? Як він реалізується?
2. Наведіть приклади типових задач обчислювальної математики. Класифікуйте їх за видом математичного апарату та за фізичною сутністю. Покажіть застосування цих методів для розв'язання прикладних задач автоматизації та управління.
3. Як визначаються похибки обчислень? Класифікуйте їх.
4. Чим відрізняються локальна та глобальна похибки?
5. Доведіть головні властивості арифметичних операцій з похибками з розділу 1.3.
6. Як визначаються ітераційні алгоритми?
7. Що таке збіжність ітераційного алгоритму?
8. Що таке стійкість ітераційного алгоритму?
9. Що таке коректність обчислювального методу?

Глава 2. Задачі лінійної алгебри

В цьому розділі розглянуто розв'язання двох найбільш поширених обчислювальних задач лінійної математики: розв'язання систем лінійних рівнянь та відшукування власних значень та власних векторів матриць. Причому вважається, що користувач вже знайомий з головними відомостями з теорії матриць.

2.1. Розв'язання систем лінійних рівнянь

У загальному випадку задача формулюється таким чином: знайти значення x_1, x_2, \dots, x_n , що задовольняють систему з n лінійних рівнянь

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= c_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= c_2, \\ \dots & \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= c_n, \end{aligned} \quad (2.1)$$

або в матричній формі $AX = C$, де

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix},$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}.$$

Необхідною і достатньою умовою існування розв'язку є нерівність нулю визначника A (лінійна незалежність рівнянь):

$$\det A \neq 0.$$

Методи розв'язання систем лінійних рівнянь можна розділити на прямі та ітераційні. До прямих, які дозволяють одержати точний розв'язок, відносяться методи визначників Крамера, Гаусса, прогонки. Ітераційні методи, що ґрунтуються на одержанні і уточненні послідовних наближень

до точного розв'язку, ефективні в тому випадку, коли є багато нульових коефіцієнтів або високий порядок системи (метод Гаусса ефективний до порядку 10^4 , ітераційні – до 10^6).

2.1.1. Прямі методи

Широко відомий метод Крамера (визначників) детально розглянутий в стандартних курсах вищої математики не може бути застосований в більшості практичних задач через велику складність розрахунку визначників навіть при невеликому зростанні порядку системи. Тому в цьому розділі зосередимося на розгляданні методу Гаусса, який, якщо й поступається ітераційним методам в певних практичних областях, все ж таки є найбільш універсальним, а також методу прогонки, що використовується в задачах з діагональними матрицями.

2.1.1.1. Метод Гаусса

Метод Гаусса (метод виключення) ґрунтується на доведенні матриці коефіцієнтів системи (2.1) до трикутного вигляду:

$$\begin{bmatrix} * & * & * & \dots & * & * \\ 0 & * & * & \dots & * & * \\ 0 & 0 & * & \dots & * & * \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & * & * \\ 0 & 0 & 0 & \dots & 0 & * \end{bmatrix}$$

і складається з двох етапів: прямого ходу і зворотного підставлення. Етап прямого ходу закінчується, коли одне з рівнянь системи стає рівнянням з одним невідомим. Далі, здійснюючи зворотне підставлення, знаходять всі невідомі. Цей метод легко алгоритмізується на ЕОМ.

Спочатку діленням коефіцієнтів на $a_{1,1}$ нормується перше рівняння, потім одержане рівняння множать на коефіцієнти $a_{1,i}$ і віднімають від усіх рівнянь. Таким чином, x_1 виключається із всіх рівнянь, окрім першого. На наступному етапі аналогічна процедура застосовується до останніх $(n-1)$ рівнянь і повторюється, доки систему не буде приведено до трикутного вигляду.

На k -му кроці нормуються коефіцієнти k -го рівняння, а нові коефіцієнти в наступних рівняннях визначаються як

$$b_{ij} = a_{ij} - a_{ik}b_{kj}, \quad i > k.$$

Коефіцієнти a_{ij} змінюються на кожному кроці.

Число арифметичних операцій для реалізації методу

$$N \approx \frac{2}{3}n^3$$

Алгоритм методу наведений на рисунку 2.1.

2.1.1.2. Метод виключення Гаусса-Жордана

Цей метод дозволяє привести матрицю коефіцієнтів до діагонального вигляду. Єдиною його формальною відмінністю від попереднього методу є те, що замість $i > k$ підставляється $i \neq k$ (k -й ряд називається провідним). В методі Гаусса перетворення зачіпали тільки рівняння, що стоять нижче провідного ряду. В методі Гаусса-Жордана перетворюються рівняння, які стоять і під провідним рядом і над ним.

Цей метод полегшує одержання розв'язку, але супроводжується збільшенням обсягу обчислень.

2.1.1.3. Модифікований метод Гаусса

У багатьох випадках виникає необхідність розв'язання систем лінійних рівнянь із матрицею коефіцієнтів, яка змінюється, і постійним стовпцем вільних членів. Найчастіше для розв'язання таких задач використовується модифікований метод Гаусса. В цьому методі матрицю коефіцієнтів A з матричного рівняння (2.1) подають у вигляді добутку лівої і правої трикутних матриць

$$L * R = A.$$

Оскільки діагональні елементи однієї з матриць дорівнюють одиниці, їх можна не запам'ятовувати, і обидві матриці зберігати в пам'яті ЕОМ на місці матриці коефіцієнтів.

У варіанті методу, що називається методом Краута, використовується така послідовність знаходження елементів матриць

$$K = 1, 2, \dots, n;$$

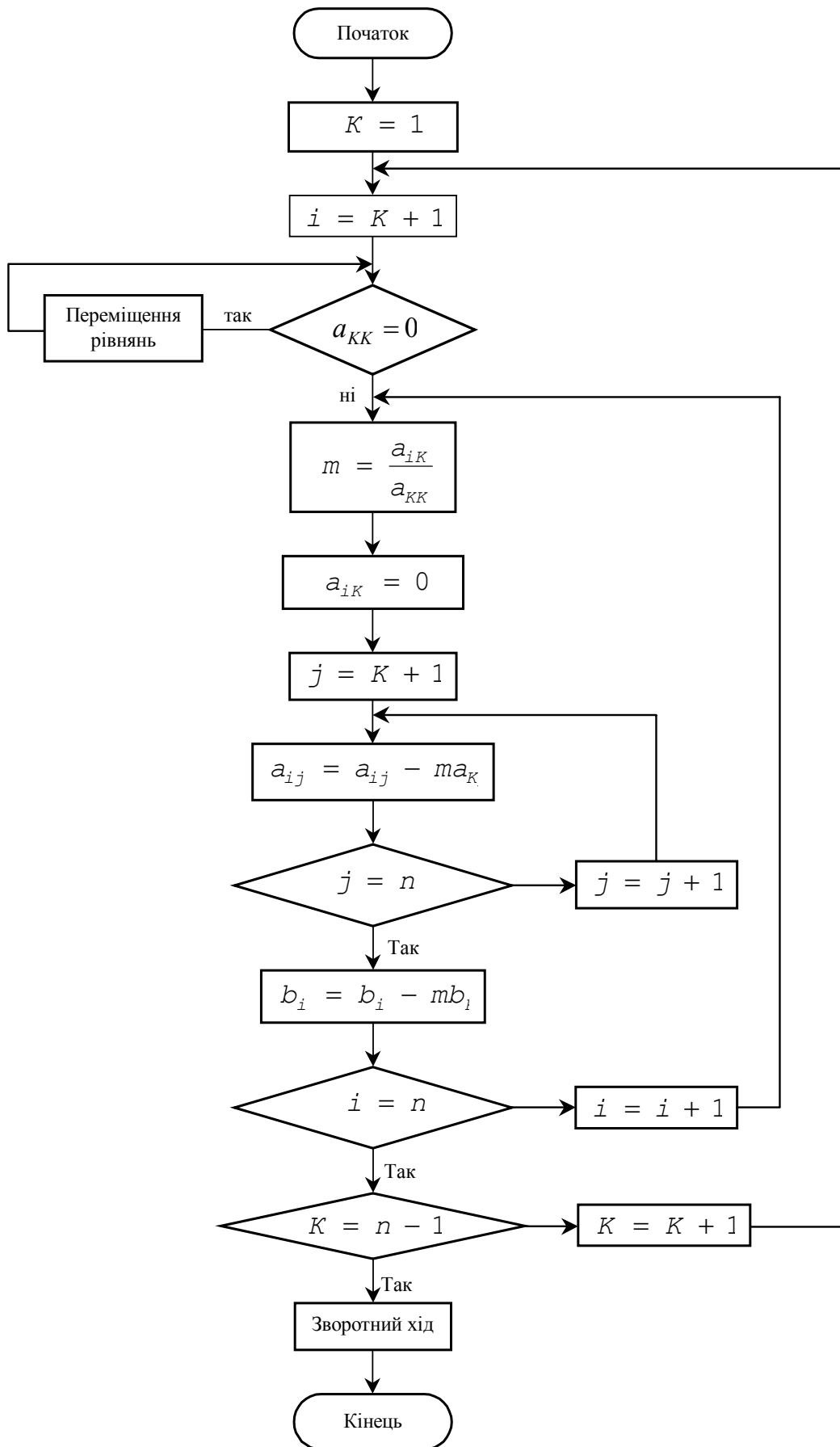


Рис. 2.1. Алгоритм методу Гаусса

$$l_{ik} = a_{ik} - \sum_{p=1}^{k-1} l_{ip} r_{pk}, \quad i = k, k+1, \dots, n;$$

$$l_{kk} = \frac{1}{l_{kk}},$$

$$r_{kj} = l_{kk} \left(a_{kj} - \sum_{p=1}^{k-1} l_{kp} r_{pj} \right), \quad j = k+1, \dots, n;$$

$$r_{kk} = 1.$$

Система $AX = C$ зводиться до системи $LRX = C$, розв'язання якої замінюється розв'язанням двох систем з трикутними матрицями:

$$LY = C,$$

$$RX = Y.$$

Елементи Y, X знаходять з таких співвідношень:

$$y_1 = l_{11} c_1;$$

$$y_i = l_{ii} \left(c_i - \sum_{p=1}^{i-1} l_{ip} y_p \right) \quad i = 2, \dots, n;$$

$$x_i = y_i - \sum_{p=i+1}^n r_{ip} x_p, \quad i = n-1, \dots, 1.$$

Число арифметичних операцій, необхідних для розв'язання цим методом системи лінійних алгебраїчних рівнянь, $N = 2n^2$.

2.1.1.4. Метод прогонки

Метод прогонки застосовується для розв'язання систем рівнянь з стрічковими матрицями коефіцієнтів. Розглянемо його застосування для розв'язання тридіагональної системи, до якої часто зводиться лінійна крайова задача.

Запишемо систему в такому вигляді:

$$\begin{aligned}
 b_0 y_0 + c_0 y_0 &= \Phi_0, \\
 a_1 y_0 + b_1 y_1 + c_1 y_2 &= \Phi_1, \\
 a_2 y_1 + b_2 y_2 + c_2 y_3 &= \Phi_2, \\
 &\dots\dots\dots \\
 a_i y_{i-1} + b_i y_i + c_i y_{i+1} &= \Phi_i, \\
 a_{n-1} y_{n-2} + b_{n-1} y_{n-1} + c_{n-1} y_n &= \Phi_{n-1}, \\
 a_n y_{n-1} + b_n y_n &= \Phi_n.
 \end{aligned}$$

Для розв'язання цієї системи виконаємо аналог прямого ходу методу Гаусса. Тоді одержимо систему у вигляді

$$\begin{aligned}
 y_0 - u_0 y_1 &= v_0, \\
 y_1 - u_1 y_2 &= v_1, \\
 &\dots\dots\dots \\
 y_{i-1} - u_{i-1} y_i &= v_{i-1}, \\
 y_i - u_i y_{i+1} &= v_i, \\
 y_{n-1} - u_{n-1} y_n &= v_{n-1}, \\
 y_n &= v_n,
 \end{aligned}$$

де $u_0, v_0, u_1, v_1, \dots, u_n, v_n$ – деякі коефіцієнти, що називаються прогонковими.

Відзначимо, що

$$u_0 = -\frac{c_0}{b_0}; \quad v_0 = \frac{\Phi_0}{b_0}.$$

Прогонкові коефіцієнти, визначені із останньої системи, дають можливість знайти y_n, y_{n-1}, \dots, y_0 .

Виключаючи із попередніх співвідношень y_{i-1} шляхом арифметичних перетворень, одержуємо формули визначення шуканих значень:

$$\begin{aligned}
 u_i &= -\frac{c_i}{a_i u_{i-1} + b_i}, \\
 v_i &= \frac{\Phi_i - a_i v_{i-1}}{a_i u_{i-1} + b_i}
 \end{aligned}$$

і далі

$$\begin{aligned}
 y_n &= v_n, \\
 y_i &= u_i y_{i+1} + v_i, \quad i = n-1, \dots, 1, 0.
 \end{aligned}$$

2.1.2. Ітераційні методи

Ітераційні методи особливо ефективні при великому порядку системи. Вони застосовуються в системах, попередньо приведених до вигляду:

$$\begin{aligned}x_1 &= b_{1,n}x_n + b_{1,n-1}x_{n-1} + \dots + b_{1,1}x_1 + b_{1,0}, \\x_2 &= b_{2,n}x_n + b_{2,n-1}x_{n-1} + \dots + b_{2,1}x_1 + b_{2,0}, \\x_n &= b_{n,n}x_n + b_{n,n-1}x_{n-1} + \dots + b_{n,1}x_1 + b_{n,0};\end{aligned}\tag{2.2}$$

або в матричній формі: $X = BX + B_0$, де

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \dots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}, \quad B_0 = \begin{bmatrix} b_{10} \\ b_{20} \\ \vdots \\ b_{n0} \end{bmatrix}.$$

Існує декілька основних різновидів ітераційних методів. Це методи Якобі (простой ітерації), Гаусса-Зейделя і послідовної верхньої релаксації, в основі яких лежить систематичне уточнення значень змінних, заданих на початку розрахунку.

В методі Якобі початкові значення змінних використовуються для обчислення нових значень x_1, x_2, \dots, x_n за допомогою наведених нижче рівнянь. Процес припиняється, коли всі нові значення виявляються достатньо близькими до початкових. В протилежному випадку нові значення використовуються замість початкових. Ця процедура повторюється доти, доки не буде досягнута збіжність або стане ясно, що процес розбіжний. В цьому методі заміна значень всіх змінних проводиться одночасно (одночасне зміщення).

Система ітераційних рівнянь має вигляд:

$$\begin{aligned}x_1^{(m+1)} &= b_{11}x_1^{(m)} + b_{12}x_2^{(m)} + b_{13}x_3^{(m)} + \dots + b_{1n}x_n^{(m)} + b_{10}, \\x_2^{(m+1)} &= b_{21}x_1^{(m)} + b_{22}x_2^{(m)} + b_{23}x_3^{(m)} + \dots + b_{2n}x_n^{(m)} + b_{20}, \\&\vdots \\x_n^{(m+1)} &= b_{n1}x_1^{(m)} + b_{n2}x_2^{(m)} + b_{n3}x_3^{(m)} + \dots + b_{nn}x_n^{(m)} + b_{n0};\end{aligned}$$

де $x_i^{(m+1)}, x_i^{(m)}$, відповідно, значення, x_i ($i = 1, \dots, n$) на наступній $(m+1)$ і попередній (m) ітераціях.

В методі Гаусса-Зейделя уточнене значення x_1 відразу ж використовується для обчислення x_2 . Потім по нових значеннях x_1 і x_2 визначаються x_3 і т.д. Це дозволяє істотно збільшити швидкість збіжності.

В методі послідовної верхньої релаксації нові значення кожної змінної обчислюються як:

$$x_i^{(m+1)} = x_i^{(m)} + \omega(\bar{x}_i^{(m+1)} - x_i^{(m)}),$$

де $\bar{x}_i^{(m+1)}$ – уточнене значення $x_i^{(m)}$ за методом Гаусса-Зейделя; ω – параметр релаксації ($1 \leq \omega \leq 2$).

При $\omega = 1$ цей метод подібний до методу Гаусса-Зейделя. Швидкість збіжності залежить від ω .

Одним з головних питань щодо застосування ітераційних методів є збіжність. Для оцінки збіжності обчислюються норми матриці коефіцієнтів $\|B\|$ з системи (2.2).

Найбільшого поширення набули такі способи оцінки норм:

$$\text{I норма: } \max_{1 \leq j \leq n} \sum_{i=1}^n |b_{ij}|,$$

$$\text{II норма: } \max_{1 \leq i \leq n} \sum_{j=1}^n |b_{ij}|,$$

$$E\text{-норма (Евклідова): } \sqrt{\sum_{i=1}^n \sum_{j=1}^n b_{ij}^2}.$$

Існує декілька підходів до визначення збіжності за допомогою оцінки норм. В загальному випадку достатньо, щоб хоча б одна з норм матриці B була менша за одиницю

$$\|B\| < 1.$$

В математиці називають таку умову “звичайною” або “сильною”. В багатьох випадках збіжність забезпечується і при виконанні так званої “слабкої” ознаки. Наприклад, “слабка” ознака сум по рядках: для всіх сум коефіцієнтів рядків ($i = 1, \dots, n$) виконується співвідношення:

$$\sum_{j=i}^n b_{ij} \leq 1,$$

але є один рядок p для якого

$$\sum_{j=i}^n b_{pj} < 1.$$

Аналогічно визначається “слабка” ознака сум по стовпцях.

“Слабка” ознака може використовуватися в тих випадках, коли матриця коефіцієнтів A системи рівнянь (2.1) є нерозкладна, тобто це квадратна матриця A , яку (на відміну від розкладної) не можна привести до вигляду:

$$\begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix},$$

де A_1, A_3 – квадратні матриці.

Для розкладних матриць система рівнянь (2.1) розпадається на дві системи рівнянь, що вирішуються послідовно. В більш детальних підручниках, список яких наведено в кінці цієї книги, міститься багато більш детальних доведень і аналіз властивостей та ознак для оцінки збіжності, але для загальноінженерного підходу до багатьох практичних задач достатньо даних, що наведені вище.

2.2. Визначення власних значень матриць

2.2.1. Постановка проблеми

Багато інженерних задач зводяться до розгляду систем рівнянь, що мають єдиний розв'язок лише в тому разі, якщо відомо значення деякого внутрішнього параметра. Цей особливий параметр називають характеристичним або власним параметром (значенням) системи. При аналізі автоматичних систем власні значення дозволяють визначити критичні параметри керувальних впливів, перевищення яких призводить до втрати стійкості системою. При динамічному аналізі механічних або радіоелектронних систем з самозбудженням власні значення відповідають резонансним частотам коливань, а власні вектори характеризують амплітуди цих коливань.

Формулювання проблеми:

Знайти n скалярних значень λ і власних векторів X , що задовольняють матричне рівняння

$$AX = \lambda X \tag{2.3}$$

Разом з пересічною проблемою власних значень існує так звана узагальнена проблема власних значень (задача для двох матриць A і F):

$$AX = \lambda FX.$$

Тут ми розглянемо тільки ту проблему, що виникає з рівняння (2.3).

Згадаємо деякі визначення з теорії матриць:

– квадратна матриця A називається особливою (сингулярною), якщо

$$\det A = 0;$$

– матриця A називається ортогональною, якщо

$$A^T A = EA,$$

тобто $A^{-1} = A^T$, де T – знак транспонування;

– матриці A і B подібні, якщо існує така несингулярна матриця P , що справедливо співвідношення

$$B = P^{-1}AP.$$

Доцільно перед аналізом методів визначення власних значень зупинитись на властивостях власних значень і векторів:

1. Всі n власних значень симетричної матриці розміру $n * n$, що складається з дійсних чисел, дійсні.

2. Якщо власні значення матриці різні, то її власні вектори ортогональні. Сукупність n лінійно незалежних власних векторів складає базис простору, що розглядається. Тому для сукупності лінійно незалежних векторів x_i ($i = 1, 2, \dots, n$) будь-який вектор в тому ж просторі можна виразити через власні вектори.

3. Якщо дві матриці подібні, то їх власні значення збігаються. З подібності A і B виходить, що $B = P^{-1}AP$, оскільки $AX = \lambda X \Rightarrow P^{-1}AX = \lambda P^{-1}X$. Якщо прийняти $X = PY$, тоді $P^{-1}APY = \lambda Y$, тобто A і B не тільки мають однакові λ , але і їх власні вектори зв'язані співвідношенням $X = PY$.

4. Добуток власного вектора матриці на скаляр, є власним вектором тієї ж матриці. Як правило, власні вектори нормалізують, розділивши їх на найбільший елемент або на суму квадратів елементів.

2.2.2. Методи обчислення власних значень

Вибір найбільш ефективного методу визначення власних значень або власних векторів для заданої задачі залежить від ряду факторів (тип рівнянь, число і характер власних значень, вид матриці і т.д.). Алгоритми визначення власних значень можна розподілити на 3 групи:

прямі, основані на розкритті характеристичних (вікових) визначників $\det(A - \lambda E) = 0$ і розв'язанні характеристичних рівнянь;

ітераційні, основані на багатократному застосуванні ітераційного алгоритму, який наближає власний вектор, що одержується в кожному циклі, до точного розв'язку;

перетворення подібності, які використовують властивості подібних матриць, що мають однакові власні значення і ортогональні власні вектори.

2.2.2.1. Прямі методи

Найбільш очевидним шляхом розв'язання задач на власні значення є їх визначення з системи рівнянь, яка має ненульовий розв'язок лише в разі, коли $\det(A - \lambda E) = 0$.

Розв'язання в цьому випадку складається з двох етапів:

– розгорнення вікового визначника безпосередньо або одним з відомих методів: Данілевського, Крилова, Леверрьє, невизначених коефіцієнтів, інтерполювання і т.д.;

– розв'язання одержаного характеристичного рівняння, корені якого і будуть власними значеннями матриці. Для визначення коренів можна використати один з методів розв'язання нелінійних алгебраїчних рівнянь (див.гл. 3).

В таблиці 2.1 наведені результати порівняння ефективності різних методів розгортання вікових визначників, де критерієм є кількість обчислювальних операцій, а в таблиці 2.2 порівняння методів розв'язання з точки зору складності, точності та швидкості збігання. Більш детальний опис цих методів можна знайти в підручниках, список яких наведено в кінці книги.

2.2.2.2. Ітераційні методи

Ітераційні методи основані на багатократному використанні ітераційного алгоритму, що наближає власний вектор, який одержується в кожному циклі, до точного розв'язку.

Таблиця 2.1

Кількість дій, що використовуються різними методами розгортання вікового визначника в залежності від його порядку.

Метод	Порядок матриці									
	3		4		5		7		9	
	Мно- жен- ня- ділен- ня (М-Д)	Дода- вання- відні- мання (Д-В)	М-Д	Д-В	М-Д	Д-В	М-Д	Д-В	М-Д	Д-В
Безпосеред- нього розгортання	12	10	60	46	320	238	13692	10078	986400	725758
Данілев- ського	14	12	42	36	92	80	282	252	632	576
Крилова	67	38	179	118	389	280	1287	1022	3209	2688
Леверрьє	41	27	153	114	414	330	1791	1533	5228	4644
Невизна- чених коефіцієнтів	67	41	171	116	364	265	1189	945	2966	2481
Інтерполяції	46	38	125	102	279	230	972	826	2525	2202

Розрахунок починається з початкового нормованого вектора $X^{(0)}$, який множиться зліва на матрицю A , і результат дорівнює добутку постійної (власне значення λ) і нормованого вектора $X^{(1)}$:

$$AX^{(0)} = \lambda X^{(1)}.$$

Якщо вектор $X^{(1)}$ збігається з $X^{(0)}$, то розрахунок припиняється. Інакше вектор $X^{(1)}$ використовується в якості початкового, і всі обчислення повторюються доти, доки $|X_i^{(n)} - X_i^{(n-1)}| \leq \varepsilon$, де $X^{(n)} \in \{x_i^{(n)}\}$ і $X^{(n-1)} \in \{x_i^{(n-1)}\}$ – вектори, що одержані в $n-1$ і n ітераційних циклах, а ε – задана припустима похибка обчислень.

Якщо процес збігається, одержаний постійний множник відповідає найбільшому власному значенню λ_{max} , а нормований вектор – відповідному власному вектору.

Аналогічно можна знайти найменше власне значення. Для цього початкова система рівнянь попередньо множиться на обернену матрицю A^{-1} :

$$A^{-1}AX = \lambda A^{-1}X,$$

звідки одержуємо $\frac{1}{\lambda} X = A^{-1}X$.

Подальше розв'язання задачі на власні значення за алгоритмом рисунку 2.1 приводить до максимального власного значення $\lambda_{max}^* = \frac{1}{\lambda_{min}}$, за яким знаходиться найменше власне значення λ_{min} .

Визначивши максимальне або мінімальне власне значення, можна знайти наступні за ним за величиною, замінивши вихідну матрицю матрицею, яка містить лише решту власних значень.

Принцип ортогональності власних векторів $X_i^T X_j = 0$ при $i \neq j$ і $X_i^T X_j = 1$ при $i = j$ дозволяє довести твердження, що якщо створити нову матрицю $A^* = A - \lambda_1 X_1 X_1^T$, то вона буде мати власне значення $\lambda_1 = 0$, а всі решта її власних значень будуть збігатися з власними значеннями вихідної матриці A . В результаті знаходимо λ_2 і далі, утворюючи A^{**} і т.д., аналогічно визначаємо всі власні значення і вектори.

2.2.2.3. Методи перетворень подібності

Методи перетворень подібності основані на властивостях подібних матриць, які мають однакові власні значення і ортогональні власні вектори.

Ці методи використовуються для одержання з вихідної матриці нової, подібної до неї, але простішого вигляду. Очевидно, найкращим спрощенням було б приведення матриці до діагонального (або блочно-діагонального) вигляду. В цьому випадку власні значення відповідали б елементам матриці, що стоять на головній діагоналі (в випадку дійсних власних значень), або просто визначались би з квадратних матриць розміру 2×2 (в випадку комплексних), що стоять на головній діагоналі. Приведення до діагонального вигляду часто важко здійснити, тому в результаті більшість методів приводить матрицю до тридіагонального вигляду, для якого існують достатньо добре досліджені і відпрацьовані на практиці методи знаходження власних значень. Як приклад можна навести метод, що приводить до послідовності Штурма і має таку властивість, що корені полінома i -го порядку розташовуються між коренями поліномів $(i+1)$ -го порядку (це значно спрощує їх визначення, оскільки обмежує ділянки відокремлення коренів). Вихідний визначник тридіагональної матриці

$$\det(A - \lambda E) = \begin{vmatrix} a_1 - \lambda & b_2 & & & \\ b_2 & a_2 - \lambda & & & \\ & & \dots & & \\ & & & a_{n-1} - \lambda & b_n \\ 0 & & & b_n & a_n - \lambda \end{vmatrix}$$

може бути представлений послідовністю поліномів Штурма

$$f_m(\lambda) = (a_m - \lambda)f_{m-1}(\lambda) - b_m^2 f_{m-2}(\lambda),$$

за умови $f_0(\lambda) = 1$ і $f_1(\lambda) = a_1 - \lambda$.

Приймаючи $m = 2, \dots, n$ і послідовно визначаючи корені, знаходимо власні значення як корені поліному n порядку.

Якщо в разі симетричних матриць ефективним способом знаходження власних значень є приведення матриці до тридіагонального вигляду, то для симетричних матриць доцільно користуватися методами перетворення подібності для приведення вихідної матриці вигляду Гессенберга:

$$\begin{vmatrix} * & 0 & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ * & * & * & * & 0 \\ * & * & * & * & * \end{vmatrix}.$$

Для приведення матриць до тридіагонального вигляду (або вигляду Гессенберга) існує ряд методів, найбільш відомими з яких є методи Якобі, Гівенса, Хаусхолдера, які докладно розглядаються в спеціальній літературі. Переваги має (з точки зору швидкодії і кількості обчислювальних операцій) метод відображень, що запропонований Хаусхолдером. Цей метод дозволяє при виконанні одного кроку перетворень перетворити в нуль одразу всі елементи цілого рядка і стовпця, що стоять зовні трьох діагоналей, в той час як в методі Гівенса на кожному k -му кроці для цього необхідно зробити $(n^2 - 3n + 2)/2$ перетворень, а в методі Якобі створення нового нульового елемента може супроводжуватись його зникненням в іншому місці матриці, тобто не гарантовано одержання результату за кінцеву кількість кроків.

Найбільш загальними з методів перетворень подібності є методи LR і QR , що основані на представленні матриці A в вигляді добутків $A = LR$ і

$A = QR$, де L – ліва трикутна матриця з одиничними діагональними елементами; R – права трикутна матриця; Q – ортогональна матриця.

Далі, застосовуючи перетворення подібності

$$A^* = L^{-1}AL = L^{-1}(LR)L = RL,$$

$$A^* = Q^TAL = Q^T(QR)Q = RQ,$$

одержуємо послідовність матриць, яка прямує до квазідіагонального вигляду.

2.2.3. Порівняння методів визначення власних значень

Основними критеріями, що дозволяють оцінити ефективність методів знаходження власних значень, стосовно їх алгоритмізації і використання для розв'язання задач на ЕОМ, є точність, швидкодія (кількість обчислювальних операцій) і необхідний розмір пам'яті ЕОМ.

Швидкодія методу залежить від двох складових: кількості операцій в єдиному циклі і кількості циклів, необхідних для одержання результату.

Важливим критерієм оцінки методів визначення власних значень є їх загальність (універсальність), тобто пристосованість до матриць різних видів, незалежно від симетрії і вигляду елементів (дійсні або комплексні).

Проаналізуємо з точки зору цих критеріїв методи знаходження власних значень.

Прямі методи приводять до необхідності розв'язання нелінійних алгебраїчних рівнянь. Методи їх розв'язання достатньо добре розроблені. Але в задачах на власні значення часто зустрічаються кратні корені, при цьому ітераційні методи не гарантують одержання розв'язку. Швидкодія цих методів різко зменшується при підвищенні порядку матриці внаслідок розкриття вікового визначника.

Ці недоліки ускладнюють використання прямих методів знаходження власних значень для великих матриць (порядку більше 10).

Істотним недоліком ітераційних методів розв'язання задач знаходження власних значень є накопичення помилок при кожному кроці ітераційного процесу (при визначенні максимального значення, створенні нової матриці і т.д.), тому практично ними можна користуватися для визначення не більше трьох-чотирьох власних значень.

В інших випадках доцільно користуватись методами перетворення подібності.

Найбільш ефективні методи перетворення подібності гарантують одержання результату при достатньо високій точності розв'язку. Часто найефективніший розв'язок задачі забезпечується при одночасному

використанні декількох методів перетворення подібності. Наприклад, методом Хаусхолдера матриця приводиться до вигляду Гессенберга, а потім методом QR визначаються власні значення.

Таким чином, вибір найбільш зручного алгоритму для розв'язання різних задач знаходження власних значень визначається типом власних значень, виглядом матриці і кількістю шуканих власних значень. Чим більш складна задача, тим менше методів та алгоритмів, з яких можна вибирати. Найбільш універсальний алгоритм QR , але він і один з найскладніших. При малих порядках матриці $n < 3...5$ можна користуватись найпростішими прямими і ітераційними методами, а при збільшенні порядку і ускладненні вигляду матриці рекомендуються лише методи перетворення подібності. Таблиця 2.2 дозволяє полегшити вибір методу розв'язання задачі на власні значення. Звичайно пакети математичного забезпечення ЕОМ містять програми, в яких використовується більшість з цих алгоритмів. Одним з ефективних способів використання наявних ресурсів ЕОМ є одночасне застосування двох підпрограм, що дозволяє поєднати їх найкращі якості.

Таблиця 2.2

Вибір алгоритму розв'язання задачі на власні значення

Назва алгоритму	Застосовується для матриць	Результат	Рекомендується для пошуку власних значень			Примітка
			найбільшого або найменшого	Всі $x < 6$	Всі $x \geq 6$	
1	2	3	4	5	6	7
Визначник (ітерація)	Загального вигляду	Власні значення		x		Потребує знаходження коренів полінома загального вигляду
Ітерація (ітерація)	Загального вигляду	Власні значення і власні вектори	x	x	x	Забезпечує найкращу точність для найбільшого і найменшого власних значень
Метод Якобі (перетворення)	Симетричних	Діагональна форма матриці		x	x	Теоретично потребує нескінченного числа кроків
Метод Гівенса (перетворення)	Симетричних	Тридіагональна форма матриці		x	x	Потребує знання коренів простого полінома
	Несиметричних	Форма Гессенберга		x	x	Потребує застосування додаткового методу
Метод Хаусхолдера (перетворення)	Симетричних	Тридіагональна форма матриці		x	x	Потребує знання коренів простого полінома

Продовження табл. 2.2

1	2	3	4	5	6	7
Метод Хаусхолдера	Несиметричних	Форма Гессенберга		x	x	Потребує застосування додаткового методу
Метод LR (перетворення)	Загального вигляду	Квазидіагональна форма матриці		x	x	Буває нестійкий
Метод QR (перетворення)	Загального вигляду	Квазидіагональна форма матриці		x	x	Найкращий метод, що має найбільшу узагальненість

Контрольні запитання та завдання

1. Чим відрізняються прямі та непрямі методи розв'язання систем лінійних рівнянь? Дати порівняльну оцінку.
2. Розкрити суть методів Крамера, Гаусса (і його різновидів), прогонки.
3. Як подається система рівнянь для застосування ітераційних методів?
4. Як перевірити збіжність ітераційних алгоритмів?
5. Чим відрізняються алгоритми ітераційних методів Якобі, Гаусса-Зейделя, послідовної верхньої релаксації?
6. Розв'язати наведену нижче систему рівнянь методом Крамера. Скласти алгоритм і програму розв'язання.

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 = 15; \\ x_1 - x_3 + 7x_4 = 26; \\ x_1 - 2x_2 + 3x_3 - 4x_4 = -10; \\ x_1 - x_2 + 2x_3 + 3x_5 = 20; \\ x_1 + x_3 - x_4 + 10x_5 = 50 \end{cases}$$

7. Розв'язати систему рівнянь з прикладу 6 методом Гаусса. Скласти алгоритм і програму.
8. Розв'язати систему рівнянь з прикладу 6 методом Гаусса-Жордана. Скласти алгоритм і програму.
9. Розв'язати систему рівнянь з прикладу 6 модифікованим методом Гаусса. Скласти алгоритм і програму.
10. Чи можна розв'язати систему рівнянь з прикладу 6 методом прогонки? Скласти алгоритм і програму для розв'язання методом прогонки системи з n рівнянь та тридіагональною матрицею коефіцієнтів.

11. Розв'язати вказаними в п. 6, 7, 8, 9, 10 методами на ЕОМ таку систему:

11.1

$$A = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 2 \\ -1 & -1 & -2 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 12 \\ 30 \\ 14 \\ 3 \end{bmatrix};$$

11.2

$$A = \begin{bmatrix} 2 & 0 & 7 & 1 \\ 1 & 2 & -1 & -1 \\ -1 & -2 & 1 & 2 \\ 0 & 2 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 11 \\ 2 \\ 0 \\ 6 \end{bmatrix};$$

11.3

$$A = \begin{bmatrix} 1 & 3 & 4 & 5 & 1 \\ 1 & 1 & 2 & -1 & 3 \\ -3 & -2 & 1 & 2 & 0 \\ 1 & 1 & 1 & 2 & 1 \\ -1 & 1 & 2 & 3 & 15 \end{bmatrix}, \quad C = \begin{bmatrix} 38 \\ 5 \\ 4 \\ 14 \\ 19 \end{bmatrix}.$$

12. Розв'язати методом Якобі наведену нижче систему. Скласти алгоритм і програму. Перевірити збіжність.

$$\begin{cases} x_1 = \frac{x_2}{2} + \frac{x_3}{3} - \frac{x_4}{4}, \\ x_2 = \frac{x_1}{10} + \frac{x_2}{10} - \frac{x_3}{10} + \frac{x_4}{2}, \\ x_3 = \frac{x_1}{5} + \frac{x_2}{5} + \frac{x_3}{3} + \frac{x_4}{10}, \\ x_4 = \frac{x_3}{3} + \frac{x_4}{4} + 2. \end{cases}$$

13. Розв'язати методом Гаусса-Зейделя систему з п. 12. Скласти алгоритм і програму. Перевірити збіжність
14. Розв'язати методом послідовної верхньої релаксації систему з п. 12 з параметром релаксації $\omega = 1,5$.
15. Сформулювати проблему визначення власних значень матриці.

16. Яка матриця називається несингулярною?
17. Сформулюйте умову ортогональності матриці.
18. Назвіть властивості власних значень та векторів.
19. Чим відрізняються прямі та ітераційні методи визначення власних значень та векторів?
20. Визначити прямим методом власні значення та вектори матриць:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 3 & 3 & 1 \end{bmatrix}; \quad A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 2 & 0 \\ 2 & 1 & 2 \end{bmatrix}.$$

21. Скласти алгоритм, програму та визначити власні значення матриць з п. 20 ітераційним методом.
22. Дати загальну характеристику методів перетворень подібності.
23. Чим відрізняються LQ та QR методи ?
24. Дати порівняльну оцінку, аналіз ефективності та умов застосування різних методів визначення власних значень та векторів.

Глава 3. Нелінійні задачі

В цьому розділі розглянуті методи розв'язання нелінійних рівнянь та систем рівнянь. До них зводиться багато практичних задач, наприклад, розрахунки нелінійних електричних кіл та систем керування, розв'язання нелінійних диференціальних рівнянь, аналіз стійкості систем шляхом оцінки їх власних значень та ін.

Якщо для найпростіших видів алгебраїчних рівнянь (не вище третього ступеня) існують точні аналітичні формули, то для трансцендентних рівнянь і будь-яких систем рівнянь таких методів взагалі не існує і слід користуватися тільки наближеними ітераційними методами та алгоритмами, найбільш поширені з яких розглянуто нижче.

3.1. Розв'язання нелінійних рівнянь

Рівняння, в які входять тільки степені аргументу з відповідними коефіцієнтами, називаються алгебраїчними.

Нелінійні рівняння, що містять тригонометричні або інші спеціальні функції, називаються трансцендентними.

Загальний вигляд алгебраїчного рівняння:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0. \quad (3.1)$$

Можна виділити деякі важливі властивості алгебраїчних рівнянь, що спрощують визначення коренів. Тут і далі ми називаємо деякі властивості теоремами, як це прийнято в математиці, але наводимо їх без доведень, щоб не ускладнювати сприйняття матеріалу. За власним бажанням можна довести ці теореми в якості додаткових самостійних завдань.

1. ОСНОВНА ТЕОРЕМА АЛГЕБРИ. Алгебраїчне рівняння порядку n має n коренів, які можуть бути як дійсними, так і комплексними.

Кожен корінь рахується відповідну кількість разів, що дорівнює його кратності. Кратність кореня x_0 рівняється k , якщо

$$f'(x_0) = f''(x_0) = \dots = f^{(k-1)}(x_0) = 0.$$

2. Якщо всі коефіцієнти a_i рівняння (3.1) дійсні, то всі комплексні корені утворюють комплексно-спряжені пари.

3. ТЕОРЕМА ДЕКАРТА. Число позитивних дійсних коренів дорівнює або менше за число змін знаків в послідовності коефіцієнтів (те

ж твердження справедливе відносно числа від'ємних дійсних коренів при заміні в (3.1) x на $-x$).

4. ТЕОРЕМА ЛАГРАНЖА. Верхня межа позитивних дійсних коренів визначається як

$$R = 1 + k \sqrt[k]{\frac{B}{a_0}}, \quad a_0 > 0,$$

де k – номер першого від'ємного коефіцієнта; B – найбільша абсолютна величина від'ємного коефіцієнта.

5. ТЕОРЕМА ГЮА. Якщо рівняння (3.1) має дійсні корені і дійсні коефіцієнти, то

$$a_k^2 > a_{k-1}a_{k+1}.$$

Нагадаємо, що прямі аналітичні методи існують лише для алгебраїчних рівнянь не вище третього порядку, а для трансцендентних рівнянь прямих методів взагалі не існує. При визначенні дійсних коренів чисельними методами треба враховувати дві теореми. Перша дозволяє відокремити корені, тобто встановити якомога тісніші інтервали $[\lambda, \beta]$, в яких знаходиться один і тільки один корінь рівняння, а друга – оцінити ступінь наближення.

ТЕОРЕМА 1. Якщо безперервна функція $f(x)$ приймає значення різних знаків на кінцях відрізка $[\lambda, \beta]$, тобто $f(\lambda)f(\beta) < 0$, то в середині цього відрізка міститься щонайменше один корінь рівняння $f(x) = 0$, тобто знайдеться хоча б одне число $\xi \in (\lambda, \beta)$ таке, що $f(\xi) = 0$.

ТЕОРЕМА 2. Нехай ξ – точний, а \bar{x} – наближений корені рівняння $f(x) = 0$, які знаходяться на одному й тому ж відрізку $[\alpha, \beta]$, причому $|f'(x)| \geq m, \quad \alpha \leq x \leq \beta$.

Тоді

$$|\bar{x} - \xi| \leq \frac{|f(\bar{x})|}{m}.$$

Існує ряд методів наближеного розв'язання нелінійних рівнянь, доцільність застосування кожного із яких визначається видом рівняння, його порядком, потрібною точністю і т.д.

Слід також пам'ятати, що при визначенні великої кількості коренів знижувати степінь початкового нелінійного рівняння шляхом ділення на $(x - x_i)$ (де x_i – корінь, що знайдено) треба дуже обережно, що пов'язано з накопиченням похибки розповсюдження, яка буде міститися в коефіцієнтах нового рівняння.

3.1.1. Метод половинного ділення

В цьому методі спочатку обчислюється значення функції в точках, що розташовані через рівні інтервали на осі x . Коли $f(x_n)$ і $f(x_{n+1})$ мають протилежні знаки, знаходять $x_{cp} = \frac{x_{n+1} + x_n}{2}$, $f(x_{cp})$. Якщо знак $f(x_{cp})$ збігається зі знаком $f(x_n)$, то надалі замість x_n використовується x_{cp} . Якщо ж $f(x_{cp})$ має знак, протилежний знакові $f(x_n)$, тобто збігається зі знаком $f(x_{n+1})$, то на x_{cp} замінюється x_{n+1} . Відзначимо, що і в цьому, і в наступних методах за умову припинення ітераційного процесу доцільно брати умову $|x_{n+1} - x_n| \leq \varepsilon$, де ε – задана похибка знаходження кореня.

Графічне зображення методу подане на рисунку 3.1, а структура алгоритму на рисунку 3.2.

Похибка розв'язку Δ через n ітерацій знаходиться в межах

$$\Delta \leq \frac{1}{2^n} |x_1 - x_0|.$$

Метод має малу швидкість збіжності, оскільки інтервал, де знаходиться корінь, з кожним кроком зменшується не більше ніж в два рази.

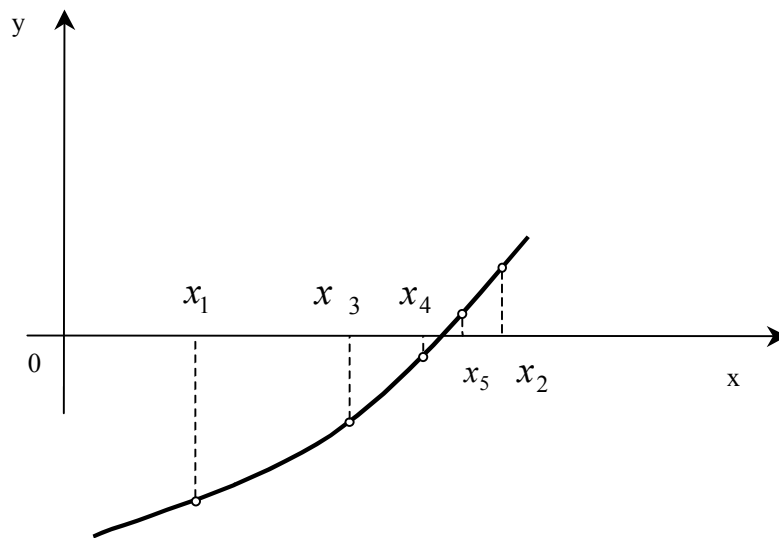


Рис. 3.1. Метод половинного ділення

3.1.2. Метод хибного положення (хорд)

Визначаються значення функції в точках, що розташовані на осі через рівні інтервали. Це робиться поки кінці інтервалів x_{n-1} , x_n не будуть мати різні знаки. Пряма, що проведена через ці дві точки, перетинає вісь у точці

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}.$$

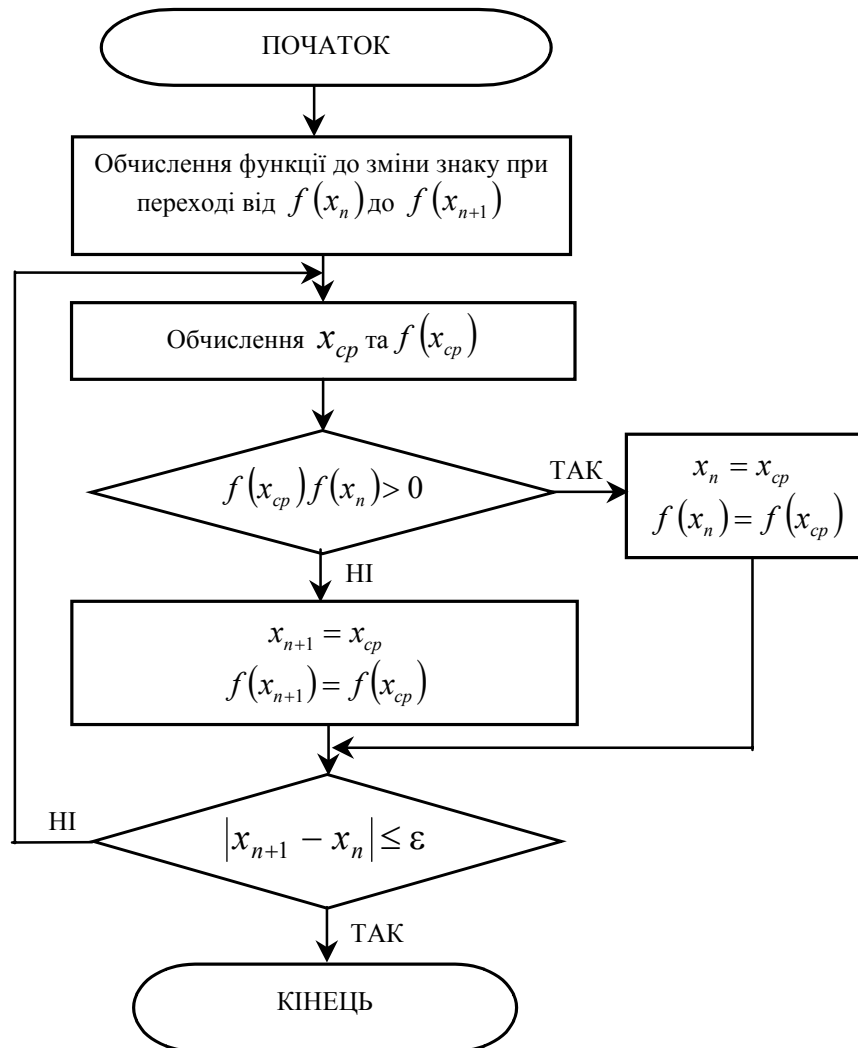


Рис. 3.2. Алгоритм методу половинного ділення

Після цього визначають $f(x_{n+1})$ і порівнюють його з $f(x_n)$. Надалі користуються x_{n+1} замість того значення, з яким воно збіглося за знаком. Якщо $|x_{n+1} - x_n| \leq \epsilon$, то вся процедура повторюється спочатку (рисунок 3.3).

Алгоритм методу хорд подібний до попереднього за винятком процедури оцінки x_{n+1} .

Треба також враховувати, що в алгоритмі обчислень за цим методом контроль похибки ведеться за тим кінцем інтервалу, що рухається. В випадку, що показаний на рисунку 3.3. аналізуються послідовні наближення: на першому кроці $|x_1 - x_2| \leq \varepsilon$, на другому – $|x_1 - x_3| \leq \varepsilon$, на третьому – $|x_3 - x_4| \leq \varepsilon$ і т. д.

Похибка розв'язку оцінюється за формулою:

$$\Delta \leq \frac{M_1 - m_1}{M_1} |x_{n+1} - x_n|,$$

де M_1, m_1 – відповідно, найбільше та найменше значення модуля першої похідної на відрізку.

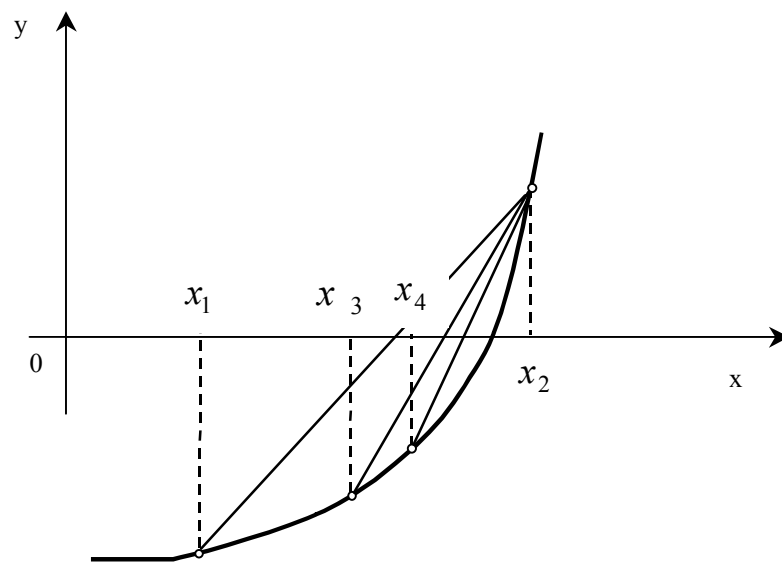


Рис. 3.3. Метод хорд

3.1.3. Метод Ньютона (дотичних)

В методі Ньютона здійснюється екстраполяція за допомогою дотичної до кривої в даній точці (рисунок 3.4)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

В основі цього методу лежить розкладання функції в ряд Тейлора

$$f(x_n + h) = f(x_n) + hf'(x_n) + \frac{h^2}{2} f''(x_n) + \dots,$$

Члени, що містять h у другому і більших степенях, відкидаються і в результаті отримується наведена вище наближена формула для оцінки x_{n+1} .

Швидкість збіжності цього алгоритму значною мірою залежить від вірного вибору початкової точки. Коли в процесі обчислень кут нахилу дотичної $f'(x)$ перетворюється на нуль, застосування цього методу ускладнюється. Можна також показати, що у випадку дуже великих значень $f''(x)$ (опуклість функції) чи кратних коренів метод Ньютона стає неефективним.

Початкове наближення слід вибирати з умови

$$(f'(x_0))^2 > f''(x_0)f(x_0) > 0.$$

Похибка методу оцінюється як:

$$\Delta \leq \frac{M_2}{2m_1} (x_{n+1} - x_n)^2,$$

де M_2 – найбільше за модулем значення другої похідної на інтервалі $[x_n, x_{n+1}]$.

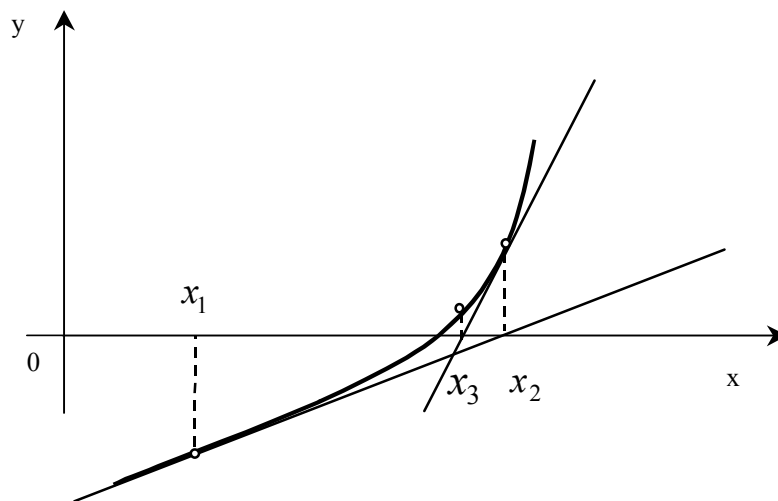


Рис. 3.4. Метод Ньютона

3.1.4. Метод січних

Однією з головних проблем при застосуванні методу Ньютона є необхідність аналітичного опису похідної. Якщо це складно чи неможливо, то можна застосувати її наближену оцінку (рисунок 3.5.) Тоді замість методу дотичних застосовується метод січних, за яким

$$x_{n+1} = x_n - \frac{f(x_n)}{F'(x_n)},$$

де $F'(x_n)$ – наближена оцінка похідної, що розглядається як січна, а не як дотична, і може бути оцінена за формулою

$$F'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

чи

$$F'(x_n) = \frac{f(x_n + h) - f(x_n)}{h},$$

де h – деякий невеликий крок.

Алгоритм цього методу подібний методу Ньютона, але з іншою ітераційною формулою.

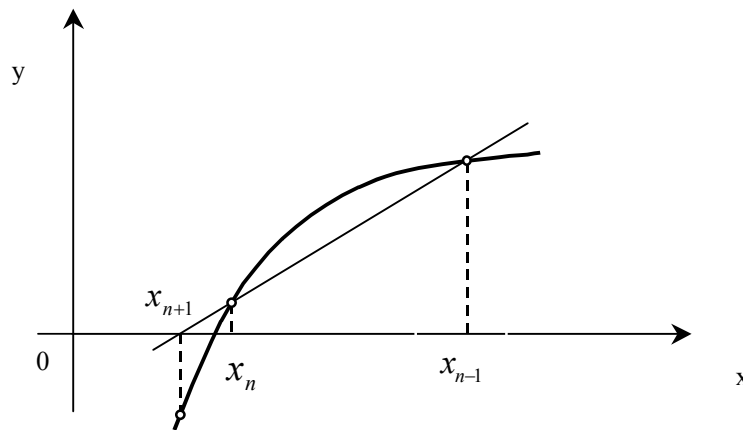


Рис. 3.5. Метод січних

3.1.5. Метод простої ітерації

Для застосування цього методу рівняння $f(x) = 0$ перетворюється до вигляду

$$x = g(x).$$

Відповідна ітераційна формула має вигляд

$$x_{n+1} = g(x_n).$$

Обчислення закінчується, коли

$$|x_{n+1} - x_n| \leq \varepsilon.$$

Для забезпечення збіжності повинна виконуватися умова про те, що максимальне за модулем значення q першої похідної функції $g(x)$ на відрізку $[x_0, x_1]$ повинно бути менше одиниці

$$q < 1.$$

Тоді процес буде збіжний незалежно від вибору початкової точки в інтервалі $x \in [x_0, x_1]$.

Похибка методу на n – й ітерації

$$\Delta \leq \frac{q^n}{1-q} |x_1 - x_0|.$$

3.1.6. Визначення комплексних коренів

Для визначення комплексних коренів можна застосовувати ті ж самі методи, що й для дійсних коренів, але при цьому оперують вже арифметикою комплексних чисел (контроль збіжності та похибки ведеться за модулем комплексного числа), що не завжди зручно для користувача.

Існує низка спеціальних методів, що дозволяють оцінювати комплексні корені, проводячи обчислення з дійсними числами. Більшість з цих методів базується на перетворенні початкового алгебраїчного рівняння (3.1) до добутку квадратичних співмножників типу

$$x^2 + px + q,$$

де p, q – коефіцієнти.

Проміжною формою для здійснення такого перетворення є рівняння у вигляді

$$(x^2 + px + q) \cdot (x^{n-2} + b_{n-1}x^{n-3} + \dots + b_3x + b_2) + b_1x + b_0 = 0, \quad (3.2)$$

де $b_1x + b_0$ – лінійний залишковий член, який прагнуть звести до нуля, і щоб початкове рівняння (3.1) ділилося на квадратичний співмножник $x^2 + px + q$ без залишку.

Для того, щоб знайти коефіцієнти $b_{n-1}, b_{n-2}, \dots, b_3, b_2$ при припущенні $b_1 = b_0 = 0$, розглянемо систему рівнянь, що виходять з еквівалентності рівнянь (3.1) та (3.2):

$$\begin{aligned}
 b_{n-1} &= a_{n-1} - p, \\
 b_{n-2} &= a_{n-2} - pb_{n-1} - q, \\
 &\vdots \\
 b_{n-j} &= a_{n-j} - pb_{n+1-j} - qb_{n+2-j}, \\
 b_3 &= a_3 - pb_4 - qb_5, \\
 b_2 &= a_2 - pb_3 - qp_4, \\
 0 &= a_1 - pb_2 - qb_3, \\
 0 &= a_0 - qb_2.
 \end{aligned}
 \tag{3.3}$$

Вона може бути розв’язана методом прогонки (2.11) чи ітераційним методом Ліна, алгоритм якого представлений на рисунку 3.6.

3.2. Розв’язання систем нелінійних рівнянь

В загальному випадку система з n нелінійних рівнянь з n невідомими подається у вигляді:

$$\begin{aligned}
 f_1(x_1, x_2, \dots, x_n) &= 0, \\
 f_2(x_1, x_2, \dots, x_n) &= 0, \\
 &\vdots \\
 f_n(x_1, x_2, \dots, x_n) &= 0.
 \end{aligned}
 \tag{3.4}$$

Оскільки нелінійні функції, що входять до системи, неможливо описати якоюсь визначеною загальною формою, то не може бути запропоновано будь-якого аналітичного прямого методу для розв’язання такої системи. З наближених ітераційних методів найбільш простим є метод простої ітерації, що базується на приведенні системи (3.4) до системи нелінійних рівнянь у вигляді:

$$\begin{aligned}
 x_1 &= g_1(x_1, x_2, \dots, x_n), \\
 x_2 &= g_2(x_1, x_2, \dots, x_n), \\
 &\dots\dots\dots \\
 x_n &= g_n(x_1, x_2, \dots, x_n).
 \end{aligned}$$

В матричному вигляді

$$X = G(X),$$

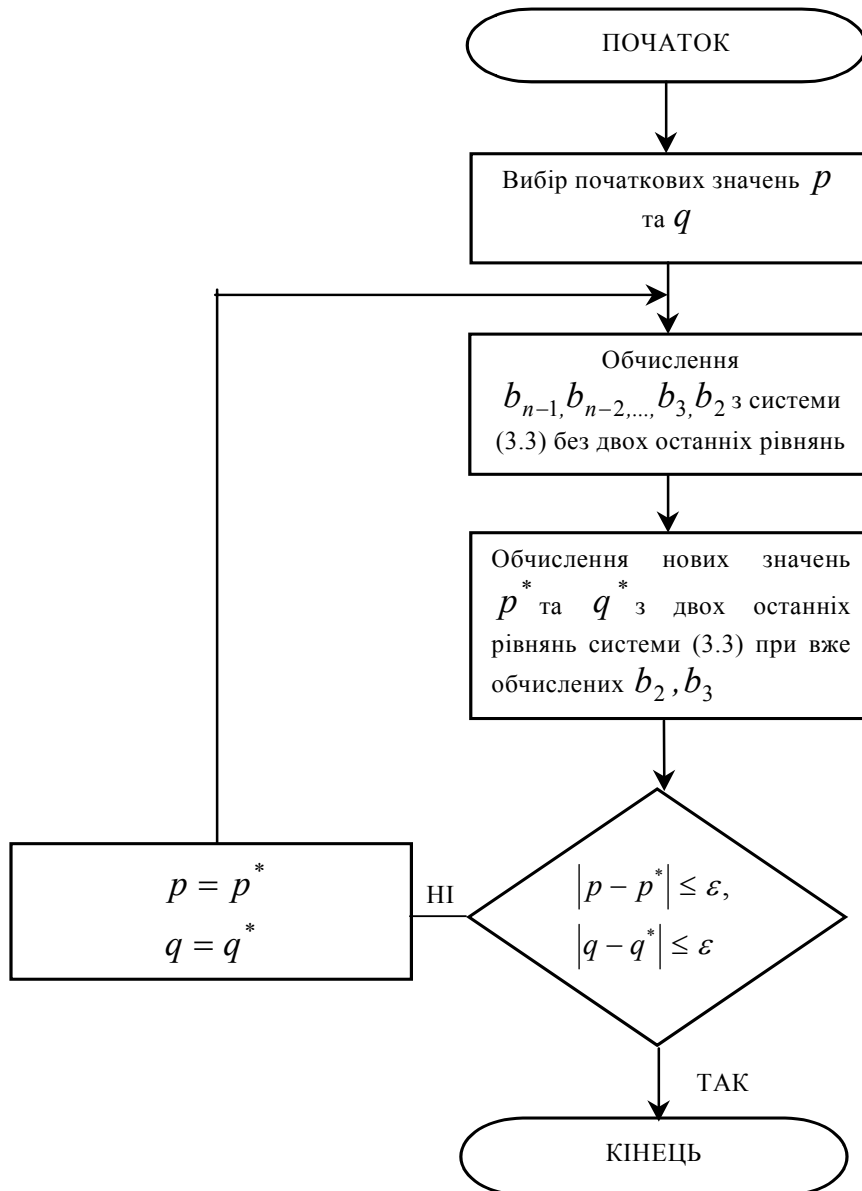


Рис. 3.6. Алгоритм методу Ліна

де

$$G(x) = \begin{bmatrix} g_1(x_1, x_2, \dots, x_n), \\ g_2(x_1, x_2, \dots, x_n), \\ \dots \\ g_n(x_1, x_2, \dots, x_n) \end{bmatrix}.$$

Далі може бути застосований алгоритм, аналогічний методу Гаусса-Зейделя для систем лінійних рівнянь. В його основі ітераційні рівняння, що пов'язують $(m + 1)$ та m ітерації

$$\begin{aligned}x_1^{(m+1)} &= g_1(x_1^{(m)}, x_2^{(m)}, \dots, x_n^{(m)}), \\x_2^{(m+1)} &= g_2(x_1^{(m+1)}, x_2^{(m)}, \dots, x_n^{(m)}), \\&\vdots \\x_n^{(m+1)} &= g_n(x_1^{(m+1)}, x_2^{(m+1)}, \dots, x_n^{(m)}).\end{aligned}$$

Для цього методу дуже важко забезпечити збіжність, а інтервал збіжності може бути настільки вузьким, що вибір початкових наближень сильно ускладнюється.

В загальному випадку цей метод буде збігатися, якщо $\|G'(x)\| < 1$, де $\|G'(x)\|$ – норма матриці частинних похідних функцій по змінних x_1, x_2, \dots, x_n .

$$G'(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \dots & \frac{\partial g_1}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \dots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}.$$

Для розв'язання систем нелінійних рівнянь широке застосування набув більш стійкий метод – метод Ньютона. Він є аналогом методу Ньютона для одного рівняння і базується на розкладанні всіх n рівнянь у ряд Тейлора:

$$f_1(x_1 + \Delta x_1, \dots, x_n + \Delta x_n) = f_1(x_1, \dots, x_n) + \Delta x_1 \frac{\partial f_1}{\partial x_1} + \dots + \Delta x_n \frac{\partial f_1}{\partial x_n} + R_n;$$

.....

$$f_n(x_1 + \Delta x_1, \dots, x_n + \Delta x_n) = f_n(x_1, \dots, x_n) + \Delta x_1 \frac{\partial f_n}{\partial x_1} + \dots + \Delta x_n \frac{\partial f_n}{\partial x_n} + R_n;$$

де R_n – члени другого та більших порядків, що в подальшому відкидаються.

Задача зводиться до розв'язання системи лінійних рівнянь

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \\ \vdots \\ -f_n \end{bmatrix}.$$

В цій системі матрицю частинних похідних називають матрицею Якобі і позначають $W(X)$.

Знайдені для певного $(m+1)$ кроку ітерації значення Δx_i використовуються як поправки до попередніх наближень

$$x_1^{(m+1)} = x_1^{(m)} + \Delta x_1,$$

.....

$$x_n^{(m+1)} = x_n^{(m)} + \Delta x_n.$$

Загальна ітераційна формула в матричному поданні має вигляд:

$$X^{(m+1)} = X^{(m)} - W^{-1}[X^{(m)}]F[X^{(m)}],$$

де $F[X^{(m)}]$ – вектор – стовпець значень функцій f_1, f_2, \dots, f_n для наближень $X^{(m)}$; $W^{-1}[X^{(m)}]$ – обернена матриця Якобі.

Алгоритм методу Ньютона подано на рисунку 3.7.

Певні труднощі при реалізації алгоритму методу Ньютона виникають при оберненні матриці Якобі. Для цього використовуються відомі з лінійної алгебри способи обернення матриць.

Існує багато варіантів застосування методу Ньютона. Наприклад, модифікований метод Ньютона

$$X^{(m+1)} = X^{(m)} - W^{-1}[X^{(0)}]F[X^{(m)}].$$

В цьому методі не треба обчислювати обернену матрицю Якобі на кожному кроці розрахунків, що спрощує алгоритм, але уповільнює збіжність і робить метод більш чутливим до вибору початкового наближення.

Метод Ньютона з параметром τ

$$X^{(m+1)} = X^{(m)} - \tau W^{-1}[X^{(m)}]F[X^{(m)}]$$

Цей метод дещо схожий з методом послідовної верхньої релаксації для систем лінійних рівнянь.

Застосовуються також різноманітні гібридні методи, в яких поєднується метод Ньютона з методом простої ітерації.

Збіжність методу Ньютона оцінюється шляхом обчислення показника

$$q = \frac{M^2 LP}{2} < 1,$$

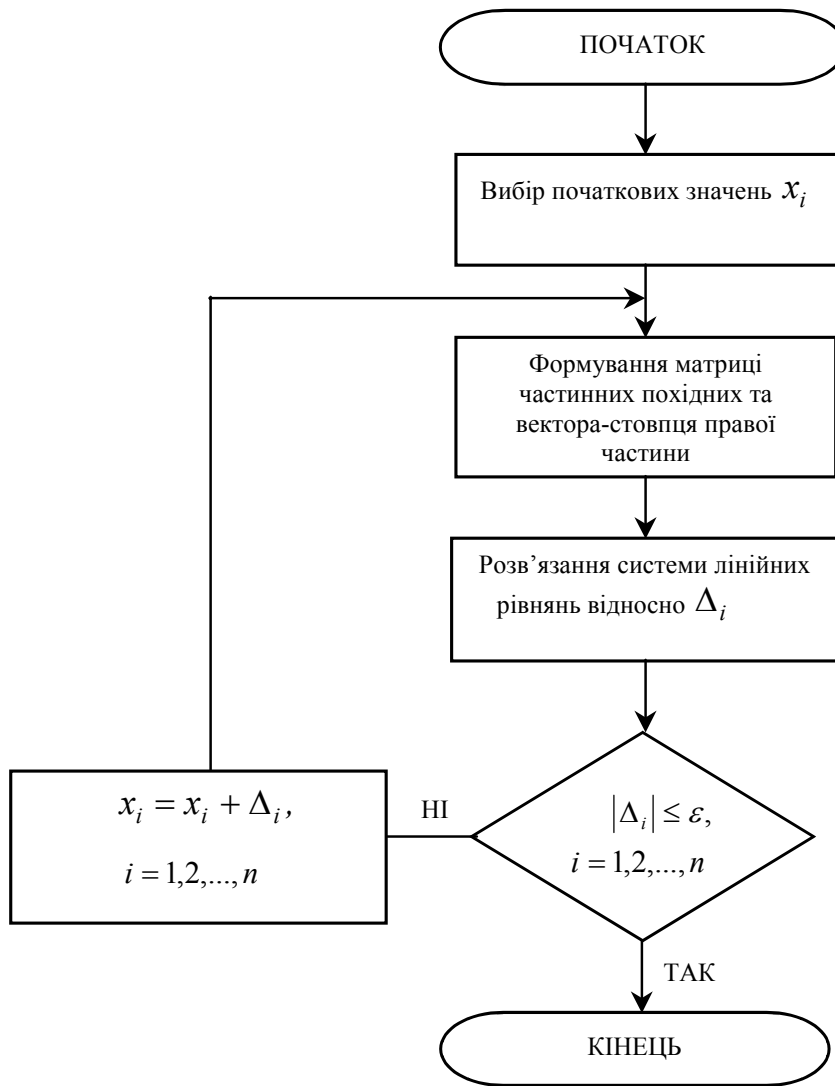


Рис. 3.7. Алгоритм методу Ньютона для систем нелінійних рівнянь

де

$$M \geq \|W^{-1}(X)\|,$$

$$L \geq \|W(X)\|,$$

$$P \geq \|F(X)\|,$$

причому

$$\lim_{l \rightarrow \infty} MP \sum_{m=0}^l q^{2^m-1} \rightarrow 0.$$

Похибка на m -ій ітерації визначається нерівністю

$$\Delta \leq MP \frac{q^{2^m-1}}{1 - q^{2^m}}.$$

Контрольні запитання та завдання

1. Навести класифікацію нелінійних рівнянь і систем та методів їх розв'язання. В яких задачах з автоматики та систем управління виникає потреба розв'язання таких задач?
2. Сформулювати головні теореми та правила відносно кількості та виду коренів алгебраїчних рівнянь.
3. Скласти алгоритм і програму та розв'язати на ЕОМ методами половинного ділення, хорд, Ньютона, січних та простої ітерації (оцінити інтервал збіжності) з похибками $\varepsilon = 0.1, 0.01, 0.001$ такі рівняння:

$$x^4 + x^2 - x + 1 = 0;$$

$$x^3 - 7x + 2 = 0;$$

$$x^5 - 10x^2 + 5 = 0;$$

$$x^3 - 2x^2 - 1 = 0;$$

$$x^5 - x^2 - 3 = 0;$$

$$x^2 - \ln x = 0;$$

$$x - 10 \sin x = 0;$$

$$x^3 - 2 \operatorname{tg} x + 5 = 0;$$

$$x^2 - x + \operatorname{tg} x = 0.$$

Порівняти ефективність (час розрахунків та кількість ітерацій для досягнення заданої похибки) різних методів.

4. Розв'язати алгебраїчне рівняння з комплексними коренями

$$x^6 + 2x^5 + 3x^4 + 4x^3 - x - 1 = 0.$$

Похибка оцінки кожного кореня не повинна перевищувати 0.05.

5. Чому для розв'язання систем нелінійних рівнянь можливо застосування тільки ітераційних методів?
6. Назвати ітераційні методи для розв'язання систем нелінійних рівнянь та порівняти їх.
7. Як оцінюється збіжність ітераційних методів розв'язання систем нелінійних рівнянь.
8. Знайти матриці Якобі для систем нелінійних рівнянь:

$$\begin{cases} x_1 + 2x_2 + x_3^2 = 14, \\ \ln|x_1| + x_2^2 - x_3 = 1, \\ x_1 + \sin \frac{\pi x_2}{4} - x_3 = 1 \end{cases}$$

та

$$\begin{cases} \cos x_1 + x_2 + \ln \frac{x_3}{3} = 2, \\ x_1 + x_2^2 + x_3 = 8, \\ \frac{\arctg x_1}{\pi} + \frac{x_2}{2} - \frac{x_3}{12} = 1. \end{cases}$$

9. Як знайти матрицю обернену заданій? Скласти алгоритм.
10. Розв'язати на ЕОМ системи рівнянь з п. 8 методом Ньютона та модифікованим методом Ньютона. Скласти алгоритм та програму. Оцінити збіжність.
11. Дати порівняльну оцінку та рекомендації щодо вибору методів розв'язання нелінійних рівнянь та систем.

Глава 4. Методи розв'язання диференціальних рівнянь

Звичайне диференціальне рівняння має нескінченну множину розв'язків. Для відшукування якого-небудь конкретного розв'язку потрібні додаткові умови. Ці умови можуть бути різними і приводити до різних задач. У випадку, коли додаткові умови задаються при одному значенні незалежної змінної, має місце задача Коші (задача з початковими умовами). Якщо умови задаються для двох або більше значень незалежної змінної, то задача стає крайовою. У задачі Коші додаткові умови називаються початковими, а у крайовій задачі – граничними. При розв'язанні цих задач використовуються різні методи і алгоритми.

Задачу Коші можна сформулювати таким чином.

Нехай дано диференціальне рівняння першого порядку

$$\frac{dy}{dx} = f(x, y). \quad (4.1)$$

Потрібно знайти функцію на відрізку від $x = a$ до $x = b$, що задовольняє як рівняння (4.1), так і початкову умову $y(a) = y_0$ (при цьому завжди припускається, що існує єдиний розв'язок на всьому відрізку).

Задача, що полягає в розв'язанні звичайного диференціального рівняння при додаткових умовах, які поставлені при декількох значеннях незалежної змінної, називається крайовою.

Крайову задачу розглянемо на прикладі звичайного диференціального рівняння другого порядку:

$$\frac{d^2 y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right)$$

при граничних умовах $y(a) = A$, $y(b) = B$.

Постановки і методи розв'язання рівнянь більш високих порядків аналогічні.

4.1. Методи розв'язання задачі Коші

Основою чисельних методів розв'язання диференціальних рівнянь слугує розкладання функції y в ряд Тейлора в околі початкової точки

$$y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{1}{2} h^2 y''(x_0) + \dots,$$

де h – відстань (крок) між початковою точкою x_0 і точкою $x_1 = x_0 + h$, в якій відшукується розв'язок.

В різних методах враховується різна кількість членів розкладання (в багатокрокових методах в поєднанні з інтерполяційними формулами), що визначає точність обчислень. При використанні цих методів на ЕОМ слід розрізняти похибки округлення через обмеженість кількості значущих цифр в ЕОМ; похибка зрізання (обмеження) – методична похибка, що пов'язана з апроксимацією розв'язків скінченними рядами, замість нескінченних, наприклад, рядами Тейлора.

Внаслідок цих причин виникають два види похибок:

локальна – сума похибок, що вносяться в обчислювальний процес на конкретному кроці;

глобальна (сумарна) – різниця між точним і обчисленим значеннями, яка включає так звану похибку розповсюдження внаслідок накопичення помилок на попередніх етапах обчислення.

Порядок методу дорівнює p , якщо існує таке позитивне число c , що

$$\Delta \leq ch^{p+1}$$

де Δ – локальна помилка; h – крок дискретизації.

Число c не залежить від номера кроку і його величини, а визначається похідними і довжиною інтервалу. При апроксимації розв'язання рядами Тейлора воно зв'язане зі степенем членів ряду, які відкидаються.

Методи розв'язання задачі Коші поділяють на однокрокові та багатокрокові.

В однокрокових методах для знаходження наступної точки на кривій $y = f(x)$ потрібна інформація лише про один попередній крок (методи Ейлера і Рунге-Кутта).

В багатокрокових методах (прогнозу і корекції) для знаходження наступної точки на кривій $y = f(x)$ потрібна інформація більш ніж про одну з попередніх точок. Щоб отримати достатньо точне чисельне значення часто використовується ітераційна процедура (наприклад, в методах Мілна-Адамса, Башфорта, Хеммінга).

4.1.1. Однокрокові методи

Найбільш простим однокроковим методом, який потребує мінімальних затрат обчислювальних ресурсів, але дає змогу обчислювати результат із порівняно низькою точністю, є метод Ейлера.

В цьому методі для оцінки наступної точки на кривій $y=f(x)$ використовується лише один лінійний член в формулі Тейлора,

$$y(x_0 + h) = y(x_0) + hy'(x_0),$$

де $y'(x_0)$ визначається з початкового рівняння.

Цей процес можна розповсюдити на наступні кроки:

$$y_{n+1} = y_n + hf(x_n, y_n).$$

Метод Ейлера є методом першого порядку ($p = 1$)

$$\Delta \leq ch^2,$$

де $c = (M_1 + M_0M_2)/2$, M_0, M_1, M_2 – визначаються як

$$\begin{aligned} M_0 &\geq |f(x, y)|, \\ M_1 &\geq \left| \frac{\partial f(x, y)}{\partial x} \right|, \\ M_2 &\geq \left| \frac{\partial f(x, y)}{\partial y} \right|, \end{aligned}$$

для всіх $x \in [a, b]$ і $y = y(x)$.

Метод Ейлера, крім значної похибки зрізання часто буває нестійким (малі локальні помилки призводять до значного збільшення глобальної).

Цей метод можна вдосконалити різними способами.

Найбільш відомі два з них: виправлений метод Ейлера і модифікований метод Ейлера (в літературі зустрічаються інші назви цих методів, наприклад, модифікований метод Ейлера й удосконалений метод ламаних).

Ітераційні формули для цих методів мають вигляд, відповідно:

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_n + h, y_n + hy_n^*)]$$

і

$$y_{n+1} = y_n + hf\left(x_n + \frac{h}{2}; y_n + \frac{h}{2}y_n^*\right),$$

де

$$y_n^* = f(x_n, y_n).$$

Геометрична інтерпретація зображена на рисунках 4.1., 4.2.

Це методи другого порядку, їх похибка має третій ступінь, що досягається покращенням апроксимації похідної. Ідея полягає у спробі

зберегти або оцінити член другого порядку у формулі Тейлора. Однак збільшення точності вимагає додаткових витрат машинного часу на обчислення y_n^* . Ще більш висока точність може бути досягнута при обчисленні вищих похідних і збереженні більшої кількості членів ряду Тейлора. Такими методами є методи Рунге-Кутта.

Принцип, на якому побудований модифікований метод Ейлера, можна пояснити, користуючись рядом Тейлора і зберігаючи в ньому член з h^2 . Апроксимація другої похідної $y''(x_0)$ здійснюється кінцевою різницею

$$y''(x_0) = \frac{\Delta y'}{\Delta x} = \frac{y'(x_0 + h) - y'(x_0)}{h}.$$

Аналогічно обчисленню другої похідної в кінцево-різницевому вигляді можна обчислити більш високі похідні: значення n -ї за значеннями попередньої $(n-1)$ -ї.

Метод Рунге-Кутта дає набір формул для обчислення координат внутрішніх точок, які потрібні для реалізації цієї ідеї. Оскільки існує ряд способів знаходження цих точок, то метод Рунге-Кутта об'єднує цілий клас методів для розв'язання диференціальних рівнянь першого порядку.

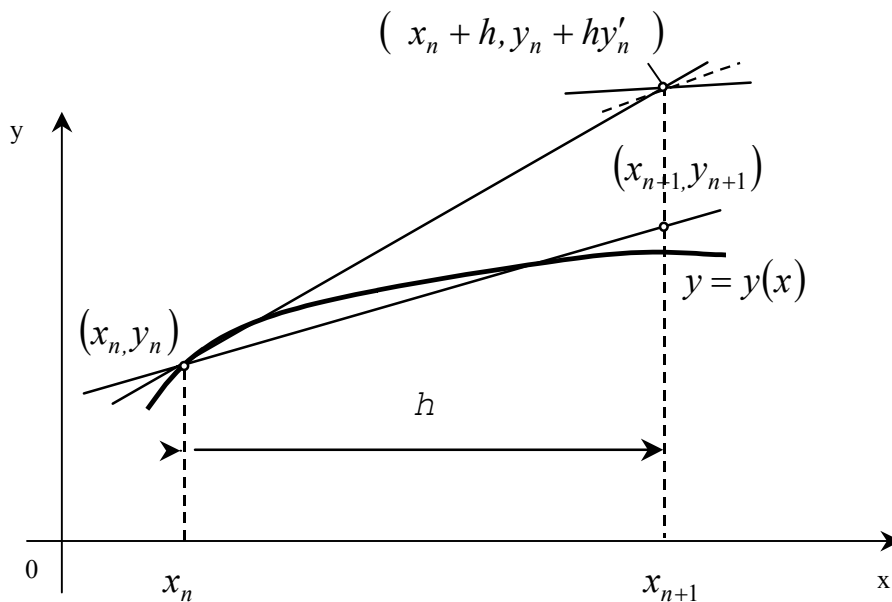


Рис. 4.1. Виправлений метод Ейлера

Найбільш розповсюджений класичний метод четвертого порядку точності:

$$y_{n+1} = y_n + \frac{k_0 + 2k_1 + 2k_2 + k_3}{6},$$

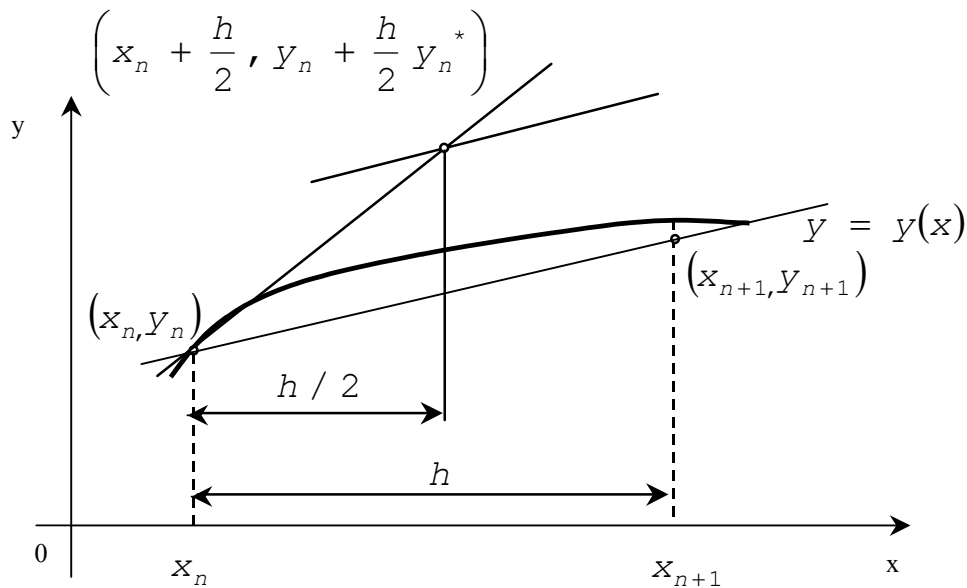


Рис.4.2. Модифікований метод Ейлера

де

$$k_0 = hf(x_n, y_n); k_1 = hf\left(x_n + \frac{h}{2}; y_n + \frac{k_0}{2}\right);$$

$$k_2 = hf\left(x_n + \frac{h}{2}; y_n + \frac{k_1}{2}\right); k_3 = hf(x_n + h; y_n + k_2).$$

Метод Ейлера і його модифікації ще називають методами Рунге-Кутта першого і другого порядку. Метод Рунге-Кутта має значно більш високу точність, що дозволяє збільшити крок розв'язання. Його максимальну величину визначає припустима похибка. Такий вибір часто здійснюється автоматично і включається як складова частина в алгоритм, побудований за методом Рунга-Кутта.

Будь-яку з формул Рунге-Кутта можна використовувати для розв'язання диференціальних рівнянь вищих порядків і систем диференціальних рівнянь. Рівняння порядку n можна звести до n диференціальних рівнянь першого порядку.

Як приклад розглянемо розв'язання звичайного диференціального рівняння другого порядку:

$$\frac{d^2 y}{dx^2} = g\left(x, y, \frac{dy}{dx}\right).$$

Нехай $z = \frac{dy}{dx}$, тоді $\frac{dz}{dx} = \frac{d^2 y}{dx^2}$;

і система набуває вигляду

$$\begin{cases} \frac{dz}{dx} = g(x, y, z), \\ \frac{dy}{dx} = f(x, y, z), \end{cases}$$

де $f(x, y, z) = z$.

Задача Коші в цьому випадку містить дві початкових умови:

$$y(x_0) = y_0; \quad z(x_0) = y'(x_0) = z_0.$$

Формули Рунге – Кутта для цього випадку мають вигляд:

$$y_{n+1} = y_n + k \quad \text{і} \quad z_{n+1} = z_n + l,$$

де $k = \frac{k_0 + 2k_1 + 2k_2 + k_3}{6}$, а $l = \frac{l_0 + 2l_1 + 2l_2 + l_3}{6}$.

Тут

$$\begin{aligned} k_0 &= h z_0, & l_0 &= h g(x_n, y_n, z_n), \\ k_1 &= h \left(z_0 + \frac{l_1}{2} \right), & l_1 &= h g \left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{l_1}{2} \right), \\ k_2 &= h \left(z_0 + \frac{l_2}{2} \right), & l_2 &= h g \left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{l_2}{2} \right), \\ k_3 &= h \left(z_0 + \frac{l_3}{2} \right), & l_3 &= h g(x_n + h, y_n + k_3, z_n + l_3). \end{aligned}$$

Раніше було відзначено, що помилка зрізання при використанні методу Рунге-Кутта n -го порядку $\Delta \leq ch^{n+1}$. Обчислення верхніх границь для коефіцієнта c являє собою складну задачу, пов'язану з необхідністю оцінки ряду додаткових параметрів. Існує декілька способів для оперативного обчислення c . Найбільшого поширення набув екстраполяційний метод Річардсона (ще його називають методом Рунге), коли послідовно знаходять значення y_n з кроком h і з кроком $\frac{h}{2}$, а після цього прирівнюють отримані величини та визначають c з рівняння:

$$y_n^{(h)} + ch^{n+1} = y_n^{\left(\frac{h}{2}\right)} + c \left(\frac{h}{2}\right)^{n+1},$$

що відповідає точному значенню y_n .

Отримаємо оціночне співвідношення:

$$c = \frac{2^{n+1}}{2^{n+1} - 1} \left[\frac{y_n^{(\frac{h}{2})} - y_n^{(h)}}{h^{n+1}} \right].$$

Можна виділити загальні риси однокрокових методів:

1. Щоб отримати інформацію у новій точці, потрібно мати дані лише в одній попередній точці. Цю властивість називають “самостартуванням”.

2. В основі всіх однокрокових методів лежить розкладання функції в ряд Тейлора, в якому зберігаються члени, що містять степені до k включно. Ціле число k називається порядком методу. Похибка на кроці має порядок $k+1$.

3. Всі однокрокові методи не вимагають обчислення похідних – обчислюється лише сама функція, але можуть вимагатися її значення в декількох проміжних точках.

4. Властивість “самостартування” дозволяє легко змінювати розмір кроку обчислення.

4.1.2. Багатокрокові методи

В цих методах для обчислення значення нової точки використовується інформація про декілька значень, що отримані раніше. Для цього використовуються дві формули: прогнозу і корекції. Алгоритм обчислення для всіх методів прогнозу і корекції однаковий та зображений на рисунку 4.3. Вказані методи відрізняються лише формулами і не мають властивості “самостартування”, оскільки вимагають знання попередніх значень. Перш ніж використовувати метод прогнозу і корекції, обчислюють початкові дані за допомогою будь-якого однокрокового методу. Часто для цього використовують метод Рунге-Кутта.

Обчислення виконують таким чином. Спочатку за формулою прогнозу та початковим значенням змінних знаходять значення $y_{n+1}^{(0)}$. Індекс (0) означає, що значення, яке прогнозується, є одним із послідовності значень y_{n+1} по мірі їх уточнення. За значенням $y_{n+1}^{(0)}$ за допомогою початкового диференціального рівняння (4.1.) знаходять похідну $y_{n+1}^{(0)'} = f(x_{n+1}, y_{n+1}^{(0)})$, яка після цього підставляється у формулу корекції для обчислення уточненого значення $y_{n+1}^{(j+1)}$. В свою чергу, за $y_{n+1}^{(j+1)}$ знаходять похідну $y_{n+1}^{(j+1)'} = f(x_{n+1}, y_{n+1}^{(j+1)})$. Якщо це значення не

достатньо близьке до попереднього, то воно вводиться у формулу корекції і ітераційний процес продовжується. У випадку близькості значень похідних визначається y_{n+1} , яке і є остаточним. Після цього процес повторюється на наступному кроці, на якому обчислюється y_{n+2} .

Зазвичай при виведенні формул прогнозу і корекції розв'язок рівняння розглядають як процес наближеного інтегрування, а самі формули отримують за допомогою методів чисельного інтегрування.

Якщо диференціальне рівняння $y' = f(x, y)$ проінтегрувати в інтервалі значень від x_n до x_{n+k} , то результат матиме вигляд

$$y(x_{n+k}) - y(x_n) = \int_{x_n}^{x_{n+k}} f(x, y) dx.$$

Цей інтеграл не можна обчислити безпосередньо, тому що $y(x)$ – невідома функція. Вибір методу наближеного інтегрування і буде визначати метод розв'язання диференціальних рівнянь. На етапі прогнозу можна використовувати будь-яку формулу чисельного інтегрування, якщо до неї не входить попереднє значення $y'(x_{n+1})$.

В таблицю 4.1 зведені найбільш розповсюджені формули прогнозу і корекції.

Для більшості методів прогнозу і корекції оцінюють похибку, користуючись таким співвідношенням:

$$\Delta \leq \frac{1}{5} [y_n^{(0)} - y_n^{(j)}]$$

Мірою похибки слугує $\epsilon \in y_n^{(j)}$, що входить до алгоритму рисунку 4.3.

Часто в довідниках приводяться більш точні формули для оцінки похибки багатокрокових методів.

При виборі величини кроку можна скористатися умовою:

$$h < \frac{2}{M_2},$$

де
$$M_2 = \left| \frac{\partial f}{\partial y} \right|_{max}.$$

Виконання цієї умови необхідно для збіжності ітераційного процесу відшукування розв'язку.

Однак у багатьох практичних випадках складність оцінки величини M_2 приводить до того, що найбільш зручним для вибору кроку є спосіб, побудований на оцінці Δ у процесі обчислень і зменшенні кроку, якщо похибка надто велика. При цьому необхідно враховувати, що оптимальне число ітерацій дорівнює двом.

Таблиця 4.1

Метод	Формула прогнозу	Формула корекції
Мілна	$y_{n+1} = y_{n-3} + \frac{4}{3}h*$ $*(2y'_n - y'_{n-1} + 2y'_{n-2}),$ $\Delta \leq \frac{28}{90}h^5 y^{(v)},$ <p>де $y^{(v)} - n''$ ята похідна $f(x, y)$</p>	$y_{n+1} = y_{n-1} + \frac{1}{3}h*$ $*(y'_{n+1} + 4y'_n + y'_{n-1}),$ $\Delta \leq \frac{1}{90}h^5 y^{(v)}$
Адамса – Башфорта	$y_{n+1} = y_n + \frac{1}{24}h*$ $*(55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}),$ $\Delta \leq \frac{251}{720}h^5 y^{(v)}$	$y_{n+1} = y_n + \frac{1}{24}h*$ $*(9y'_{n+1} - 19y'_n - 5y'_{n-1} + y'_{n-2}),$ $\Delta \leq \frac{19}{720}h^5 y^{(v)}$
Хеммінга	$y_{n+1}^{(0)} = y_{n-3} + \frac{3}{4}h*$ $*(2y'_n - y'_{n-1} + 2y'_{n-2})$ <p>Уточнення прогнозу</p> $\bar{y}_{n+1}^{(0)} = y_{n+1}^{(0)} +$ $+ \frac{112}{121}(y_n - y_n^{(0)})*$ $*[\bar{y}_{n+1}^{(0)}] = f(x_{n+1}, \bar{y}_{n+1}^{(0)})$	$y_{n+1}^{(j+1)} = \frac{1}{8}*(9y_n - y_{n-2}) +$ $+ \frac{3}{8}h*([\bar{y}_{n+1}^{(j)}] + 2y'_n - y'_{n-1})$
	$y_{n+1} = y_{n-3} + \frac{4}{3}h*$ $*(2y'_n - y'_{n-1} + 2y'_{n-2}),$ $\Delta \leq \frac{28}{90}h^5 y^{(v)}$	$y_{n+1} = \frac{1}{8}*$ $*[9y_n - y_{n-2} + 3h(y'_{n+1} + 2y'_n - y'_{n-1})],$ $\Delta \leq \frac{1}{40}h^5 y^{(v)}$

Основні особливості, що притаманні багатокроковим методам:

- 1) за допомогою цих методів не можна розпочати розв'язання задачі, оскільки для їх використання необхідна інформація про значення функції в кількох точках;
- 2) можна отримати оцінку похибки зрізання, не звертаючись до обчислення додаткових величин;

3) методи прогнозу і корекції не дозволяють легко змінювати крок обчислень, для цього необхідно весь ітераційний процес починати спочатку.

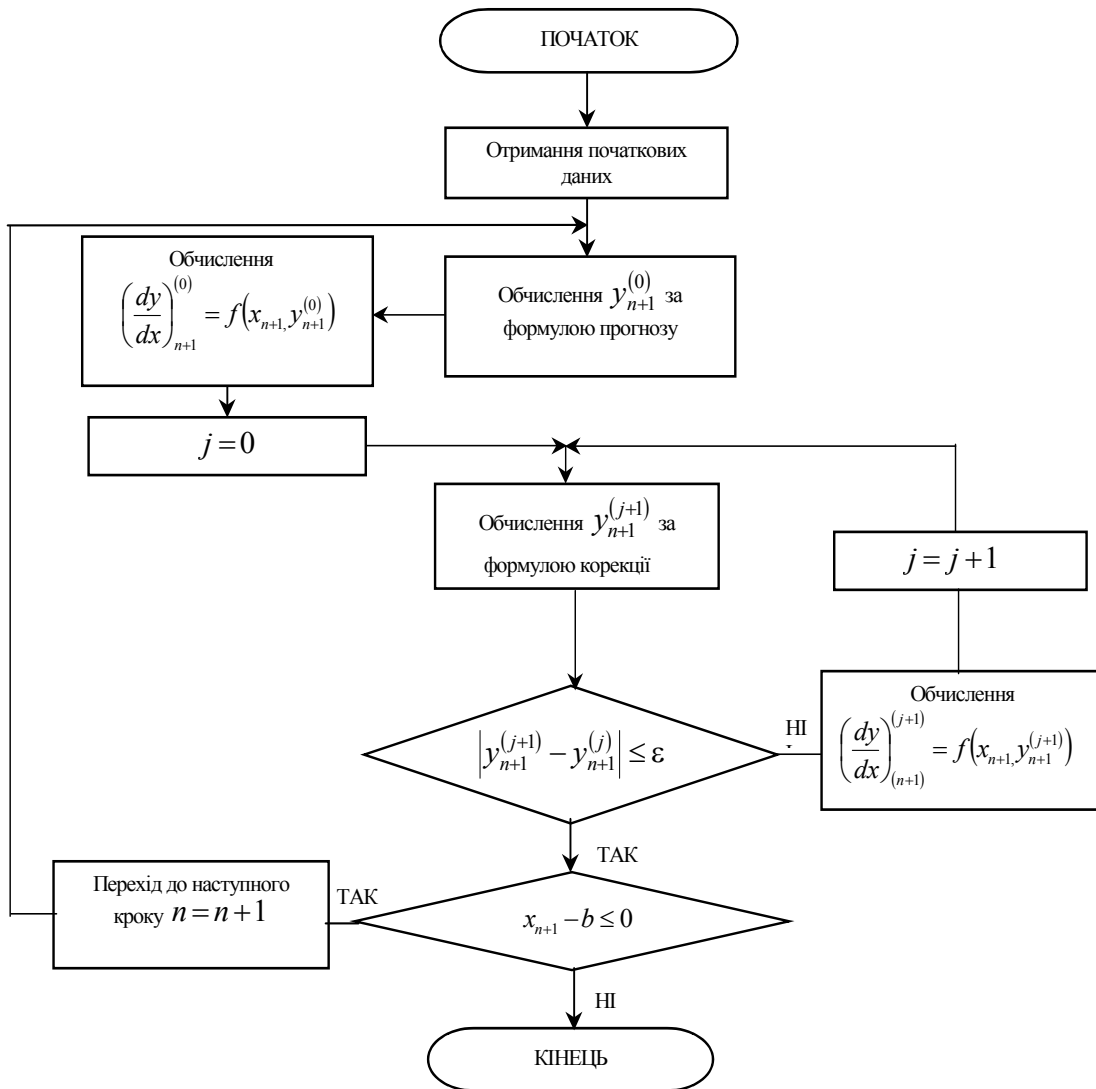


Рис. 4.3. Алгоритм методів прогнозу та корекції

4.1.3. “Жорсткі” задачі

Існують звичайні диференціальні рівняння, для яких складно отримати задовільний розв'язок з використанням описаних вище методів. Виникнення таких задач зв'язано з поняттям сталої часу диференціального рівняння, проміжком часу, коли змінна частина розв'язку зменшується в ϵ раз. Рівняння порядку n має n постійних часу; якщо будь-які дві з них сильно (на практиці в сто і більше разів) різняться за величиною або яка-небудь з них достатньо мала порівняно з інтервалом часу, на якому відшукується розв'язок, то задача називається “жорсткою” і її практично

неможливо розв'язати звичайними чисельними методами. В таких випадках крок повинен бути достатньо малим, щоб можна було враховувати приріст складових розв'язку, які найбільш швидко змінюються, навіть після того, як їх внесок стане практично непомітним. Але зменшення кроку призводить до збільшення витрат часу ЕОМ і накопиченню помилок. Розроблені спеціальні методи для розв'язування таких задач, які часто зустрічаються в теорії автоматичного керування.

Найбільш простим є так званий обернений метод Ейлера, в якому розв'язок знаходиться з рівняння, яке містить y_{n+1} у неявному вигляді:

$$y_{n+1} = y_n + hf(x_n, y_{n+1}).$$

На практиці “жорсткими” можуть виявитися рівняння, у яких коефіцієнти при похідних сильно (в сто і більше разів) відрізняються один від одного.

4.1.4. Вибір методу розв'язання задачі Коші

Порівнюючи ефективність однокрокових і багатокрокових методів, виділяють такі особливості:

1. Багатокрокові методи вимагають більшого об'єму пам'яті ЕОМ, тому що оперують більшою кількістю початкових даних.

2. При використанні багатокрокових методів існує можливість оцінки похибки на кроці, тому значення кроку обирається оптимальним, а в однокрокових – з деяким запасом, що знижує швидкодію.

3. При однаковій точності багатокрокові методи вимагають меншого обсягу обчислень. Наприклад, в методі Рунге-Кутта четвертого порядку точності доводиться обчислювати чотири значення функції на кожному кроці, а для забезпечення збіжності методу прогнозу і корекції того ж порядку точності – достатньо двох.

4. Однокрокові методи на відміну від багатокрокових дозволяють одразу почати розв'язання задачі (“самостартування”) і легко змінювати крок в процесі обчислень.

Перед початком розв'язання задачі необхідно провести перевірку на “жорсткість” і у випадку позитивного результату використати спеціальні методи. Якщо задача Коші дуже складна, то зазвичай перевага надається методу прогнозу і корекції, який має до того ж більш високу швидкодію. Початок розв'язання задачі при цьому проводиться за допомогою однокрокових методів. Якщо для обчислення чергового значення y_i вимагається більш ніж дві ітерації або якщо помилка зрізання дуже велика, то необхідно зменшити крок h . З іншого боку при дуже малій похибці зрізання можна збільшити крок, тим самим підвищити швидкодію, але при

цьому весь процес розв'язання треба починати спочатку. Інколи на практиці вимагається мінімізувати час підготовки задачі до розв'язання. Тоді доцільно використовувати методи Рунге-Кутта.

На закінчення слід відзначити, що велике значення для ефективного розв'язання задачі мають досвід, інтуїція і кваліфікація користувача як при постановці задачі, так і в процесі вибору методу розробки алгоритму і програми для ЕОМ. При цьому часто зручно користуватись вже готовими програмними засобами, які є в наявності (наприклад, в пакетах MAPLE, МАТНЕМАТИКА).

4.2. Методи розв'язання крайових задач

Методи розв'язання крайових задач розглядаються на прикладі звичайного диференціального рівняння другого порядку

$$\frac{d^2 y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right)$$

при граничних умовах $y(a) = A$, $y(b) = B$. Методи розв'язання крайових задач розділяють на дві групи: методи, що побудовані на заміні розв'язання крайової задачі розв'язанням декількох задач Коші (методи “стрілянини”) та різницеві методи.

4.2.1. Метод “стрілянини”

Якщо звичайне диференціальне рівняння другого порядку – лінійне, то воно має вигляд:

$$y'' = f_1(x)y' + f_2(x)y + f_3(x)$$

при $y(a) = A$, $y(b) = B$.

Крайову задачу можна звести до задачі Коші введенням додаткової початкової умови, крім $y(a)=A$ вводиться $y'(a)=\alpha_1$.

Знайшовши розв'язок $y_1(x)$, можна поставити іншу початкову умову $y'(a)=\alpha_2$ і отримати інший розв'язок $y_2(x)$. Якщо $y_1(b)=\beta_1$ а $y_2(b)=\beta_2$, причому $\beta_1 \neq \beta_2$, то розв'язок:

$$y(x) = \frac{1}{\beta_1 - \beta_2} [(B - \beta_2)y_1(x) + (\beta_1 - B)y_2(x)]$$

буде задовольняти обидві початкові умови.

При розв'язуванні нелінійного звичайного диференціального рівняння методами “стрілянини” крайова задача зводиться до розв'язування декількох задач Коші, послідовно вводячи в початкові умови значення α :

$$y(a)=A \text{ і } y'(a)=\alpha$$

і намагаючись знайти розв'язок, який задовольняє умову $y(b)=B$.

При цьому алгоритм досягнення мети будується на основі одного з методів оптимізації. Однак цей шлях розв'язання задачі пов'язаний з великими обчислювальними труднощами, і тому у випадку нелінійних диференціальних рівнянь перевага надається різницеvim методам.

4.2.2. Різницеvi методи

Апарат різницеvих методів являє собою потужний засіб чисельного розв'язування звичайних диференціальних рівнянь і диференціальних рівнянь у частинних похідних. У його основі лежить подання незалежного аргументу на відрізьку $[a, b]$ у вигляді дискретної множини точок x_i , $i=0, \dots, n$, $x_0=a$, $x_n=b$, яка називається сіткою.

Найбільше поширення отримала рівномірна сітка з кроком $x_i - x_{i-1} = h$. При цьому замість безперервної функції $f(x)$ розглядається сіткова функція $y_i = f(x_i)$. Аналогічно проводиться дискретизація функції багатьох змінних, наприклад, двох :

$$x_{ij}, \quad i=0, \dots, n, \quad j=0, \dots, m, \quad y_{ij} = f(x_{ij}).$$

Крім найбільш розповсюдженої прямокутної сітки використовують полярну, трикутну, скошену та інші, зображені на рисунку 4.4. Багатовимірні сітки знаходять використання в задачах з частинними похідними по декількох незалежних змінних.

Розв'язок задачі різницеvими методами складається з двох етапів:

- отримання дискретної (різницеvої) апроксимації диференціальних рівнянь і дослідження отриманих при цьому різницеvих рівнянь;
- розв'язання різницеvих рівнянь.

При отриманні різницеvих схем важливу роль відіграє загальна вимога: щоб різницева схема якомога краще наближала основні властивості початкового диференціального рівняння. Такі різницеvi схеми можна отримати за допомогою варіаційних принципів та інтегральних співвідношень. Оцінка точності різницеvої схеми зводиться до вивчення похибки апроксимації та стійкості. Сіткову функцію можна розглядати як функцію цілочислового аргументу

$$y(i) = y_i, \quad i = 0, \pm 1, \pm 2, \dots$$

Для y_i можна ввести операції, які є дискретним (різницеvim) аналогом операцій диференціювання та інтегрування.

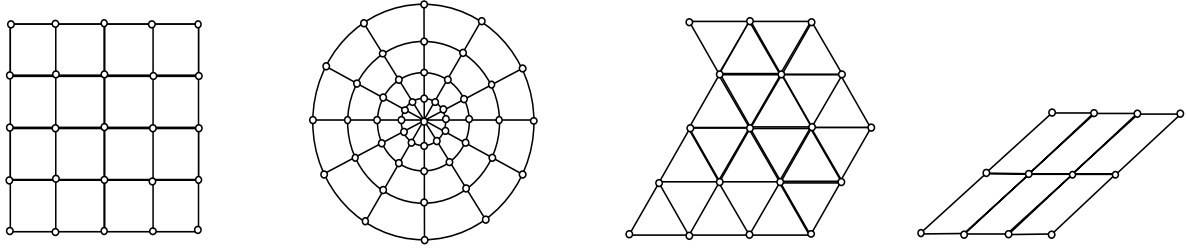


Рис.4.4. Двовимірні сітки

Аналогом першої похідної є різниці першого порядку :

$$\Delta y_i = y_{i+1} - y_i \quad - \quad \text{права різниця};$$

$$\nabla y_i = y_i - y_{i-1} \quad - \quad \text{ліва різниця};$$

$$\delta y_i = \frac{1}{2}(\Delta y_i + \nabla y_i) = \frac{1}{2}(y_{i+1} - y_{i-1}) \quad - \quad \text{центральна різниця}.$$

Зауважимо, що $\Delta y_i = \nabla y_{i+1}$.

Далі можна записати різниці другого порядку

$$\Delta^2 y_i = \Delta(\Delta y_i) = \Delta(y_{i+1} - y_i) = y_{i+2} - 2y_{i+1} + y_i \quad ,$$

$$\Delta \nabla y_i = \Delta(y_i - y_{i-1}) = (y_{i+1} - y_i) - (y_i - y_{i-1}) = y_{i+1} - 2y_i + y_{i-1} \quad ,$$

тобто $\Delta^2 y_i = \Delta \nabla y_{i+1}$.

Аналогічно визначається різниця m -го порядку:

$$\Delta^m y_i = \Delta(\nabla^{-1} y_i).$$

Звідси виходить, що

$$\sum_{j=k}^i \Delta y_j = y_{i+1} - y_k \quad , \quad \sum_{j=k}^i \nabla y_j = y_i - y_{k-1}.$$

На множині вузлів сітки, яка називається шаблоном (тут розглядаються одновимірні шаблони, в наступному розділі вводяться у

розгляд двовимірні), замінимо неперервний диференціальний оператор Ly різницеvim оператором $L_h y$.

Наприклад, різницеві оператори для першої похідної на трьох вузлах сітки $(x-h, x, x+h)$:

$$L_h^+ y = \frac{y(x+h) - y(x)}{h} = y_x^+$$

$$L_h^- y = \frac{y(x) - y(x-h)}{h} = y_x^-$$

$$L_h^0 y = \frac{y(x+h) - y(x-h)}{2h} = y_x^0$$

права, ліва і центральна, відповідно, різницеві похідні.

Аналогічно для другої похідної

$$\begin{aligned} L_h y &= \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} = \frac{y_x^+(x) - y_x^-(x)}{h} = \\ &= \frac{y_x^-(x+h) - y_x^-(x)}{h} = y_{xx}^-(x). \end{aligned}$$

При розв'язуванні крайової задачі записуються різницеві рівняння для всіх n вузлів області змінення $x \in [a, b]$. Враховуючи дві граничні умови $y_0 = y(a)$ та $y_n = y(b)$, отримують систему з $n-1$ алгебраїчних рівнянь з $n-1$ невідомими y_i . Якщо початкове звичайне диференціальне рівняння лінійне, то задача зводиться до розв'язування системи лінійних алгебраїчних рівнянь, а якщо нелінійне – то нелінійних або трансцендентних алгебраїчних систем. Привести крайову задачу, яку можна розв'язати методом кінцевих різниць, до вигляду, зручного для розробки стандартної програми для ЕОМ складно, оскільки формулювання кожної задачі залежить від вигляду диференціального рівняння, яке розглядають.

В цьому розділі поданий лише короткий опис різницевого підходу без деталізації методів побудови і дослідження різницевих схем. Взагалі різницеві методи є універсальними методами чисельного аналізу, хоча й розглядаються в більшості випадків у зв'язку з розв'язанням крайових задач і диференціальних рівнянь в частинних похідних. Наприклад, метод Ейлера розв'язання задачі Коші може бути інтерпретований як використання одновимірного різницевого оператора. Прикладом узагальненого підходу до розв'язання різноманітних задач обчислювальної математики різницеvими методами можуть слугувати праці академіка А. А. Самарського, що містять детальний аналіз і дослідження питань теорії і практики використання різницевих схем.

Контрольні запитання та завдання

1. Сформулюйте задачу Коші і крайову задачу. В чому відмінність у постановці цих задач?
2. Наведіть приклади звичайних диференціальних рівнянь, які розв'язуються лише чисельними методами.
3. Дайте порівняльну оцінку і визначте області застосування методів розв'язання звичайних диференціальних рівнянь.
4. Чи можлива крайова задача для рівнянь першого порядку ?
5. Що таке похибка зрізання (обмеження)? Чим визначається порядок похибки методу ?
6. Покажіть, що загальна формула для методу Рунге-Кутта, застосована до рівняння $y' = -y$ з початковою умовою $y(0) = 1$, приводить до співвідношення

$$y_n = \left(1 - h + \frac{h^2}{2}\right)^n \approx e^{-x_n}.$$

7. В чому подібність між методами прогнозу і корекції і виправленим методом Ейлера?
8. Як оцінюються похибки методів Ейлера, Рунге-Кутта, прогнозу і корекції?
9. Дайте геометричну інтерпретацію методів Ейлера.
10. Що таке властивість “самостартування” методів розв'язання звичайних диференціальних рівнянь.?
11. Як обирається крок при використанні багатокрокових методів розв'язання?
12. Які рівняння називають “жорсткими”? У чому містяться особливості їх розв'язання?
13. Для розв'язання яких крайових задач може використовуватись метод “стрілянини”? Як отримати розв'язок крайової задачі з розв'язків декількох задач Коші в цьому методі?
14. З яких етапів складається розв'язання задачі різницевиими методами?
15. Запишіть всі можливі різницеві оператори для першої і другої похідних.
16. Як визначаються різницеві оператори для похідних більш високих порядків?
17. З яких міркувань вибирається метод розв'язання звичайних диференціальних рівнянь?

Глава 5. Методи розв'язання диференціальних рівнянь в частинних похідних

З розв'язанням диференціальних рівнянь в частинних похідних інженерам і дослідникам доводиться зустрічатися у багатьох областях науки і техніки, в аеро- і гідродинаміці, ядерній фізиці, радіозв'язку. Математичні моделі з диференціальними рівняннями в частинних похідних широко використовуються в теорії автоматичного керування і у вимірювальній техніці. В таких рівняннях містяться частинні похідні і шукана величина залежить відразу від декількох змінних. Розглянемо диференціальне рівняння другого порядку з двома незалежними змінними:

$$\begin{aligned}
 &A(x, y) \frac{\partial^2 f}{\partial x^2} + B(x, y) \frac{\partial^2 f}{\partial x \partial y} + C(x, y) \frac{\partial^2 f}{\partial y^2} + \\
 &E(x, y) \frac{\partial f}{\partial y} + D(x, y) \frac{\partial f}{\partial x} + F(x, y) = G(x, y).
 \end{aligned}
 \tag{5.1}$$

Аналогічно звичайним диференціальним рівнянням єдиний розв'язок рівняння (5.1) можна отримати лише задавши додаткові умови, але оскільки тут присутні дві незалежні змінні x та y , умова повинна задаватися для якої-небудь кривої у площині xy . Ця умова може бути накладена на функцію f або (та) на її похідні та залежати від типу рівняння, яке визначає її вигляд і характер зміни.

Існують три типи диференціальних рівнянь другого порядку:

- еліптичні, при $B^2 - 4AC < 0$;
- параболічні, при $B^2 - 4AC = 0$;
- гіперболічні, при $B^2 - 4AC > 0$.

Рівняння можуть переходити з одного вигляду в інший в залежності від значень коефіцієнтів.

Еліптичні рівняння описують усталені (стаціонарні) процеси, причому задача ставиться в замкненій області і в кожній точці границі цієї області задаються граничні умови. Інші два типи рівнянь описують еволюційні процеси. В таких задачах найбільш поширений випадок, коли на одній частині границі ставлять граничні умови, а на іншій – початкові.

Приклади деяких диференціальних рівнянь в частинних похідних, які описують різні типи задач, наведені в таблиці 5.1.

Таблиця 5.1

Диференціальні рівняння в частинних похідних

Рівняння	Математична форма	Приклади задач, в яких зустрічається рівняння
Лапласа	$\Delta f = 0$	Усталена течія рідини. Стаціонарні теплові поля
Пуассона	$\Delta f = -R$	Теплопередача з внутрішнім джерелом тепла
Дифузії	$\Delta f = \frac{1}{h^2} \frac{\partial^2 f}{\partial t^2}$	Нестаціонарна теплопровідність
Хвильове	$\Delta f = \frac{1}{c^2} \frac{\partial^2 f}{\partial t^2}$	Розповсюдження хвиль
Бігармонічне	$\Delta^2 f = F(x, y)$	Деформація пластин

В таблиці використані прийняті позначення найбільш поширених операторів:

оператор Лапласа $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$;

бігармонічний оператор $\Delta^2 f = \frac{\partial^4 f}{\partial x^4} + 2 \frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4}$.

Існують два методи розв'язання диференціальних рівнянь в частинних похідних: різницевий метод (метод скінченних різниць) і метод скінченних елементів. В сучасній прикладній математиці обидва методи розглядаються як інтерпретації використання загальної теорії різницевого схем до розв'язання диференціальних рівнянь в частинних похідних.

В основі методу скінченних елементів лежить варіаційне обчислення. Диференціальне рівняння, яке описує задачу й відповідні граничні умови, використовується для постановки варіаційної задачі. В методі скінченних елементів фізична задача замінюється кусково-гладкою моделлю. Цей метод вимагає складної постановки задачі, високої кваліфікації й досвіду, неуніверсальний (кожне розв'язання застосовується лише для конкретної задачі). Метод скінченних елементів знайшов широке використання для розв'язання спеціальних задач в теоретичній механіці, гідродинаміці, теорії поля, він складний, вимагає серйозної підготовки і знань в конкретній області використання, та для його достатньо повного викладу прийшлося б написати спеціальний підручник. Тому в даній роботі не наводиться докладний виклад методу скінченних елементів, тим більше, що при розв'язанні задач автоматички та систем керування частіше використовується різницевий метод.

5.1. Різницьвий метод

У підрозділі 4.1. розглянуто основні означення й правила побудови різницьвих схем. Для диференціальних рівнянь другого порядку в частинних похідних найчастіше використовується двовимірний прямокутний сітка. Центральні-різницьві шаблони, які застосовують на двовимірній квадратній сітці з кроком h , зображеній на рисунку 5.1 (індекс j надається незалежній змінній y , а i відноситься до x), можуть бути отримані аналогічно одновимірному випадку в підрозділі 4.2.2.

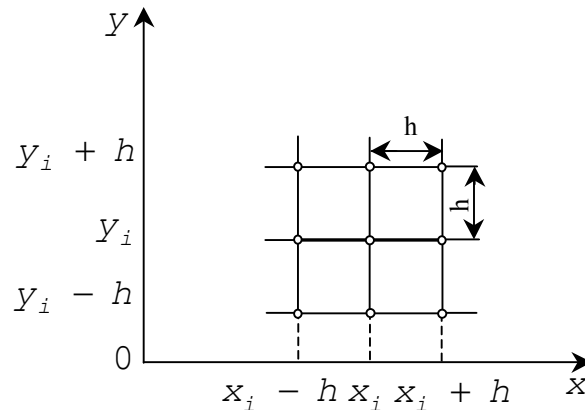


Рис. 5.1. Квадратна сітка

Для зручності позначення $f(x_i + h, y_i)$ замінимо на $f_{i+1,j}$. Користуючись цим позначенням, отримаємо вирази для частинних похідних, з якими доводиться зустрічатися на практиці, й використання яких ілюструється відповідними обчислювальними шаблонами (рисунку 5.2.):

$$\frac{\partial f}{\partial x} \approx \frac{f_{i+1,j} - f_{i-1,j}}{2h} \approx \frac{1}{2h} \quad \left[\begin{array}{ccc} (-1) & 0 & 1 \\ & & i,j \end{array} \right]$$

$$\frac{\partial f}{\partial y} \approx \frac{f_{i+1,j} - f_{i-1,j}}{2h} \approx \frac{1}{2h} \quad \left[\begin{array}{c} 1 \\ 0 \\ -1 \\ & i,j \end{array} \right]$$

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2} \approx \frac{1}{h^2} \quad \left[\begin{array}{ccc} 1 & -2 & 1 \\ & & i,j \end{array} \right]$$

Рис. 5.2. Обчислювальні шаблони для похідних

$$\frac{\partial^2 f}{\partial y^2} \approx \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h^2} \approx \frac{1}{h^2} \left[\begin{array}{c} \textcircled{1} \\ \textcircled{-2} \\ \textcircled{1} \end{array} \right]_{i,j}$$

$$\frac{\partial^2 f}{\partial x \partial y} \approx \frac{f_{i+1,j+1} - f_{i-1,j+1} - f_{i+1,j-1} + f_{i-1,j-1}}{4h^2} \approx \frac{1}{4h^2} \left[\begin{array}{ccc} \textcircled{-1} & \textcircled{0} & \textcircled{1} \\ \textcircled{0} & \textcircled{0} & \textcircled{0} \\ \textcircled{1} & \textcircled{0} & \textcircled{-1} \end{array} \right]$$

$$\frac{\partial^4 f}{\partial x^4} \approx \frac{1}{h^4} \left[\textcircled{1} - \textcircled{4} - \textcircled{6} - \textcircled{4} - \textcircled{1} \right]_{i,j}$$

$$\frac{\partial^4 f}{\partial y^4} \approx \frac{1}{h^4} \left[\begin{array}{c} \textcircled{1} \\ \textcircled{-4} \\ \textcircled{6} \\ \textcircled{-4} \\ \textcircled{1} \end{array} \right]_{i,j}$$

Продовження рис. 5.2.
Обчислювальні шаблони для похідних

З цих елементів будуються більш складні обчислювальні шаблони для диференціальних рівнянь. Додавання похідних здійснюється суперпозицією відповідних обчислювальних шаблонів. Цим методом конструюються шаблони для Δf і $\Delta^2 f$ (рисунок 5.3).

Всі наведені обчислювальні шаблони мають похибку другого порядку. Можна побудувати більш точні обчислювальні шаблони, якщо включити у розгляд додаткові вузли. Іноді, щоб звести до мінімуму розповсюдження похибок, користуються лівими або правими різницями.

Часто труднощі, які пов'язані з використанням прямокутної сітки, виникають через границю неправильної конфігурації, яка не проходить через вузли сітки. Розглянемо приклад розв'язання такої задачі для обчислювального шаблону рівняння Лапласа в області, що обмежена довільною кривою, яка зображена на рисунок 5.4.

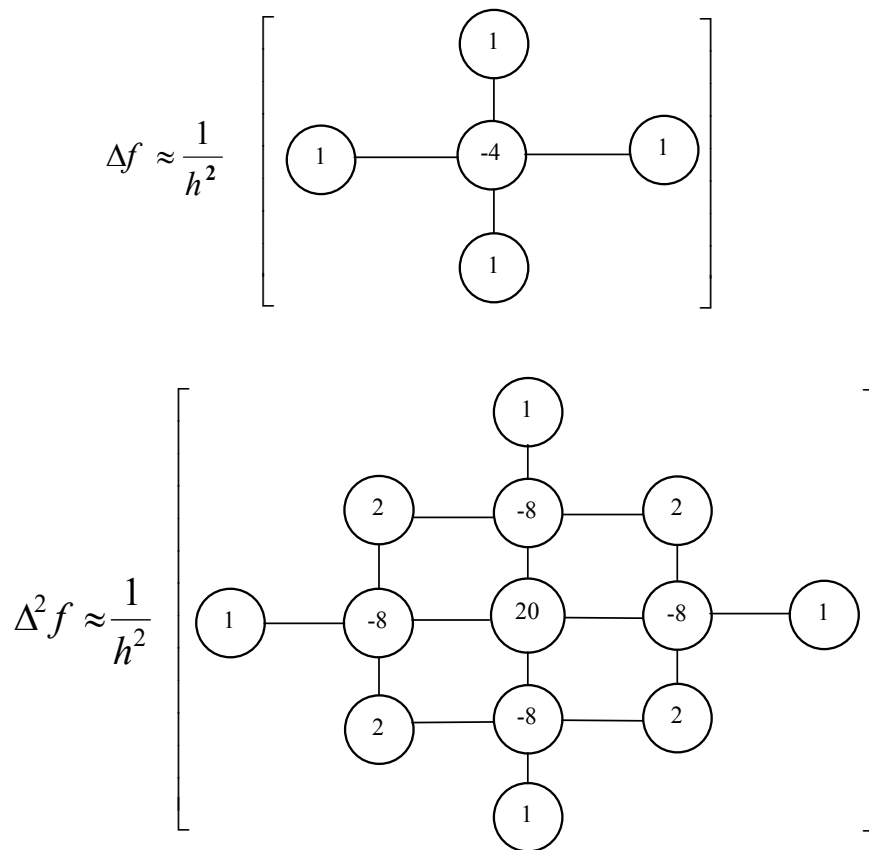


Рис. 5.3. Обчислювальні шаблони для операторів

Другі частинні похідні для вузлів, які лежать на границі області, можна записати у вигляді:

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{\frac{f_a - f_{i,j}}{ah} - \frac{f_{i,j} - f_{i-1,j}}{h}}{0,5(ah + h)};$$

$$\frac{\partial^2 f}{\partial y^2} \approx \frac{\frac{f_b - f_{i,j}}{bh} - \frac{f_{i,j} - f_{i,j-1}}{h}}{0,5(bh + h)}.$$

Додавши другі похідні, отримаємо :

$$\Delta f \approx \frac{2}{h^2} \left(\frac{f_{i-1,j}}{1+a} + \frac{f_a}{a(1+a)} + \frac{f_b}{b(1+b)} + \frac{f_{i,j-1}}{1+b} - \frac{a+b}{ab} f_{i,j} \right).$$

Застосувавши обчислювальний шаблон до кожного з n вузлів сітки, отримаємо систему з n рівнянь, яка може бути лінійною, якщо початкове

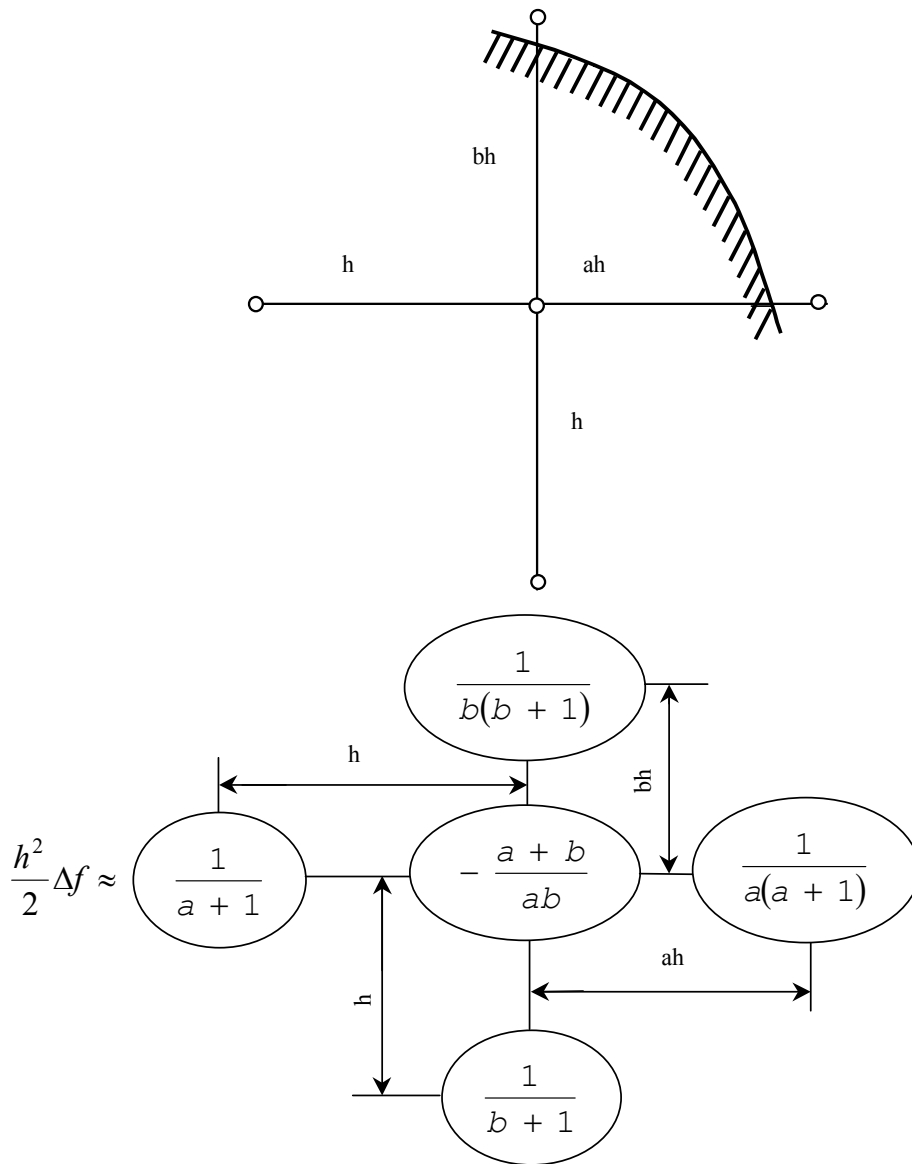


Рис. 5.4. Обчислювальний шаблон для границі неправильної форми

диференціальне рівняння має відповідну структуру. В цьому випадку розв'язання задачі зводиться до розв'язання системи рівнянь вигляду:

$$\begin{bmatrix} \text{матриця} \\ \text{коефіцієнтів} \end{bmatrix} \begin{bmatrix} \text{невідоме значення } u \\ \text{вузлах (вектор - стовпець)} \end{bmatrix} = \begin{bmatrix} \text{вектор - стовпець} \\ \text{вільних членів} \end{bmatrix},$$

яка розв'язується найчастіше ітераційними методами.

5.2. Розв'язання різних типів диференціальних рівнянь в частинних похідних

Практичні методи і алгоритми розв'язання різних виглядів диференціальних рівнянь в частинних похідних мають свої особливості і вимагають окремого розгляду на прикладі найбільш розповсюджених задач.

5.2.1. Еліптичні рівняння

До еліптичних рівнянь приводиться багато різних фізичних задач: розрахунок напружень, які виникають при пружному скруті довгого циліндричного стрижня; розподіл електричних напруг на площині, що проводить струм; задача про стаціонарні течії тепла в двовимірному тілі. Часто виникає необхідність розв'язання таких задач і в теорії автоматичного керування. Більшість еліптичних рівнянь описується рівнянням Пуассона або його окремим випадком – рівнянням Лапласа.

Розглянемо класичну задачу Діріхле для рівняння Лапласа в прямокутній області, яка формулюється таким чином: знайти неперервну функцію $f(x,y)$, яка задовольняє $\Omega = \{(x,y) / 0 \leq x \leq a, 0 \leq y \leq b\}$ всередині прямокутної області рівняння Лапласа:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

і приймає на границі області задані значення:

$$\begin{aligned} x=0; & f(0,y) = f_1(y), \\ x=a; & f(a,y) = f_2(y), \\ y=0; & f(x,0) = f_3(x), \\ y=b; & f(x,b) = f_4(x). \end{aligned}$$

Введемо в області розв'язання двовимірну сітку з кроком h по осі x і l по осі y . Тоді, користуючись прийнятими в попередніх розділах позначеннями і апроксимуючи рівняння Лапласа різницеvim рівнянням, отримаємо таку систему лінійних рівнянь (прийmemo для спрощення $l=h$):

$$\begin{aligned} f_{ij} &= \frac{1}{4}(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}), \\ f_{i,0} &= f_3(x_i), \quad f_{i,m} = f_4(x_i), \quad f_{0,j} = f_1(y_j), \quad f_{n,j} = f_2(y_j) \end{aligned} \quad (5.2)$$

при $i=1,2,\dots,n-1$; $j=1,\dots,m-1$.

Ця система рівнянь має велику кількість нульових елементів і задовольняє умови збіжності при використанні ітераційних методів. Найбільше використання для розв'язання таких систем знайшов метод Гаусса-Зейделя, який, коли застосовується до еліптичних різницевого рівнянь, називається методом Лібмана або методом послідовних зміщень. Порядок ітерацій можна простежити, переписавши систему (5.2) у вигляді:

$$\begin{aligned}
 f_{1,1}^{(m+1)} &= \frac{1}{4} [f_3(h) + f_1(h) + f_{2,1}^{(m)} + f_{1,2}^{(m)}], \\
 f_{2,1}^{(m+1)} &= \frac{1}{4} [f_3(2h) + f_{1,1}^{(m+1)} + f_{3,1}^{(m)} + f_{2,2}^{(m)}], \\
 f_{3,1}^{(m+1)} &= \frac{1}{4} [f_3(3h) + f_{2,1}^{(m+1)} + f_{4,1}^{(m)} + f_{3,2}^{(m)}], \\
 &\dots\dots\dots \\
 f_{n-1,1}^{(m+1)} &= \frac{1}{4} \{f_3[(n-1)h] + f_2(h) + f_{n-2,1}^{(m+1)} + f_{n-1,2}^{(m)}\}, \\
 f_{1,2}^{(m+1)} &= \frac{1}{4} [f_1(2h) + f_{1,1}^{(m+1)} + f_{2,2}^{(m)} + f_{1,3}^{(m)}], \\
 f_{2,2}^{(m+1)} &= \frac{1}{4} [f_{2,1}^{(m+1)} + f_{1,2}^{(m+1)} + f_{3,2}^{(m)} + f_{2,3}^{(m)}], \\
 &\dots\dots\dots
 \end{aligned}$$

де верхніми індексами позначено порядковий номер ітерації: m – попередня, $m+1$ – наступна.

Зазвичай вважають $f_{i,j}^{(0)} = 0$ для всіх i, j . Система рівнянь легко розв'язується на ЕОМ. Взагалі кажучи, будь-які еліптичні рівняння, які не містять $\frac{\partial^2 f}{\partial x \partial y}$, зводяться до систем різницевого рівнянь, які можна

розв'язувати як методом Лібмана, так і іншими ітераційними методами (Якобі, послідовної верхньої релаксації та ін.), оскільки для них

виконуються умови збіжності. Для еліптичних рівнянь, які містять $\frac{\partial^2 f}{\partial x \partial y}$ в

загальному вигляді, питання про збіжність ітераційних методів не має теоретичного розв'язку і необхідно розглядати отриману систему рівнянь в кожному конкретному випадку.

5.2.2. Гіперболічні рівняння

Найчастіше зустрічається в інженерній практиці гіперболічне рівняння в частинних похідних – хвильове рівняння, яке описує різні види коливань: коливання струни або мембрани, розповсюдження звукових хвиль в різних середовищах тощо.

В загальному вигляді задача формулюється таким чином: знайти функцію $f(x, t)$, яка задовольняє всередині області $\Omega = \{(x, t), 0 \leq x \leq a, 0 \leq t \leq T\}$ рівняння

$$\frac{\partial^2 f}{\partial t^2} = c^2 \frac{\partial^2 f}{\partial x^2} \quad (c = \text{const} > 0),$$

початкові

$$f(x, 0) = f_0(x) \quad ,$$

$$\left. \frac{\partial f}{\partial t} \right|_{x,0} = g(x)$$

і граничні умови

$$f(0, t) = \mu_1(t) \quad ,$$

$$f(a, t) = \mu_2(t) \quad .$$

Оскільки заміна змінних $t = c \tau$ приводить рівняння до вигляду:

$$\frac{\partial^2 f}{\partial \tau^2} = \frac{\partial^2 f}{\partial x^2},$$

то надалі приймаємо $c=1$.

Переходячи до різницевого рівняння на сітці з кроком h по x й τ по t з центральними різницями, отримаємо

$$\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2} = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\tau^2}.$$

Якщо ввести $r = \frac{\tau}{h}$, то вираз для $f_{i,j+1}$ прийме вигляд:

$$f_{i,j+1} = r^2 (f_{i+1,j} + f_{i-1,j}) + 2(1 - r^2) f_{i,j} - f_{i,j-1}. \quad (5.3)$$

Схема розв'язання за рівнянням (5.3) називається тришаровою, оскільки зв'язує значення $f_{i,j}$ на трьох часових шарах $j-1, j, j+1$. Ця схема явна й дозволяє в явному вигляді виразити $f_{i,j}$ через значення f з

попередніх шарів (існують неявні схеми, основані на використанні інших обчислювальних шаблонів, але вони вимагають більшого обсягу обчислень при розв'язанні системи рівнянь). Для знаходження розв'язку на першому шарі зазвичай використовують інтерполяційні методи. Наприклад,

$$f_{i,1} = f_{i,0} + \tau g(x_i). \quad (5.4)$$

Співвідношення сторін сітки визначається величиною r , яка є мірою стійкості отриманого розв'язку. При $r > 1$ розв'язок нестійкий, при $r < 1$ хоча й стійкий, але точність його при зменшенні r спадає, при $r = 1$ різницевий розв'язок стійкий і збігається з точним. Вибір $r = 1$ зручний ще й тим, що при цьому спрощується співвідношення (5.4)

$$f_{i,j+1} = f_{i+1,j} + f_{i-1,j} - f_{i,j-1}.$$

5.2.3. Параболічні рівняння

Прикладом задачі, яка приводить до параболічного рівняння в частинних похідних, є задача про теплопередачу по довгому стрижню. Вона описується рівнянням теплопередачі (або дифузії).

Задача полягає у знаходженні $f(x, t)$, яка задовольняє в області $\Omega = \{(x, t) \mid 0 \leq x \leq a, 0 \leq t \leq T\}$ рівняння

$$\frac{\partial f}{\partial t} = k \frac{\partial^2 f}{\partial x^2} \quad (k = \text{const} > 0),$$

початкові

$$f(x, 0) = f_0(x)$$

і граничні умови першого роду

$$f(0, t) = \mu_1(t) \quad ,$$

$$f(a, t) = \mu_2(t) \quad .$$

Заміна змінних $t = k t$ приводить рівняння до вигляду

$$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2},$$

тому надалі будемо вважати $k = 1$.

Можливі два варіанти отримання різницевого рівняння на сітці з кроком h по x та τ по t (рисунок 5.5).

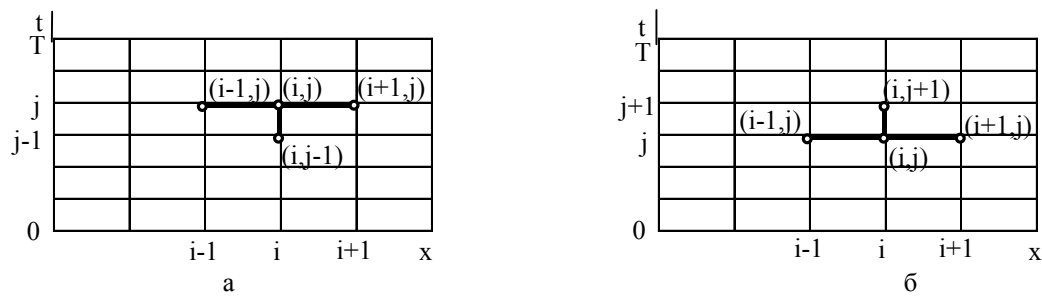


Рис. 5.5. Обчислювальні шаблони для параболічних рівнянь

Варіант з апроксимацією на чотириточковому шаблоні (рисунок 5.5, а) приводить до неявної двошарової різницевої схеми

$$2f_{i+1,j} - (1 + 2r)f_{i,j} + 2f_{i-1,j} = -f_{i,j-1},$$

де $r = \frac{\tau}{h^2}$.

Ця схема доповнена рівняннями, отриманими з крайових умов

$$f_{0,j} = \mu_1(t_j); \quad f_{n,j} = \mu_2(t_j),$$

зводить задачу до розв'язання до системи рівнянь, які мають стійкий розв'язок при будь-яких значеннях r .

Варіант з апроксимацією на чотириточковому шаблоні (рисунок 5.5, б) призводить до явної двошарової системи

$$f_{i,j+1} = rf_{i+1,j} + (1 - 2r)f_{i,j} + 2f_{i-1,j}$$

Ця схема стійка тільки при $r \leq 0,5$, що приводить до необхідності проводити обчислення з дуже малим кроком по t , який обмежує швидкодію і вимагає більших витрат часу ЕОМ. Тому для параболічних рівнянь більш широке розповсюдження отримала неявна схема.

5.3. Загальні рекомендації до розв'язання диференціальних рівнянь в частинних похідних

На початковому етапі вибирається метод розв'язання задачі. Зазвичай простіше використовувати різницевий метод, який вимагає більш простої підготовки задачі до розв'язання, але в ряді випадків для задач з

добре розробленою теорією (наприклад, задач механіки) доцільно звертатися до методу скінченних елементів.

При визначенні кроку розв'язання задачі основним фактором є точність (якщо висока, то необхідна або дуже дрібна сітка, або розбиття на дуже малі елементи). При цьому необхідно враховувати, що похибка різницевих методів має другий порядок. Її оцінку можна проводити аналогічно звичайним диференціальним рівнянням за методом Рунге.

У випадку симетрії в області розв'язання можна число вузлів зменшити в два або навіть у чотири рази (при симетрії по обох осях координат). Це дозволяє отримати економію часу та обсягу обчислень.

Велику роль для ефективного розв'язання задачі грає вибір початкових наближень. При використанні ітераційних методів від цього значною мірою залежить швидкість збігання. Часто має сенс розв'язувати задачу в декілька етапів: на першому за допомогою грубої сітки (або розбиття на крупні елементи) отримують добре початкове наближення, на наступних – шукають більш точний розв'язок на дрібній сітці.

Контрольні запитання та завдання

1. Наведіть приклади інженерних задач, які описуються диференціальними рівняннями в частинних похідних.
2. На які види можна поділити диференціальні рівняння другого порядку в частинних похідних в залежності від коефіцієнтів, які входять до них?
3. Які методи існують для розв'язання диференціальних рівнянь в частинних похідних?
4. Як конструюються обчислювальні шаблони для частинних похідних? Отримайте обчислювальні шаблони для оператора Лапласа та бігармонічного оператора.
5. Як використовується різницевий метод у випадку, коли границя області, де шукають розв'язок, не проходить через вузли сітки? Наведіть приклад обчислювального шаблону?
6. Якими різницевиими методами розв'язується класична задача Діріхле для рівнянь Лапласа в прямокутній області? Доведіть їх збіжність.
7. Отримайте тришарову схему розв'язання гіперболічних рівнянь. В яких випадках розв'язок стійкий?
8. Порівняйте ефективність використання явної і неявної схем розв'язання параболічних рівнянь в частинних похідних.
9. Як вибирається крок при розв'язанні диференціальних рівнянь в частинних похідних?

Глава 6. Методи обробки експериментальних даних

В цьому розділі розглянуті найбільш важливі задачі обробки даних (інтерполяція, апроксимація, статистична обробка), які зустрічаються в задачах проектування комп'ютеризованих систем управління.

6.1. Інтерполяція

Мета інтерполяції – побудування функції $F(x)$, яка приймає в окремих точках $x_i[a, b]$ ($i = 0, 1, 2, \dots, n$) (вузли інтерполяції) значення,

$$F(x_0) = y_0, F(x_1) = y_1, \dots, F(x_i) = y_i, \dots, F(x_n) = y_n, \quad (6.1)$$

що збігаються з раніше заданими значеннями в цих точках невідомої функції $y = f(x)$. Геометрично це означає, що потрібно знайти криву $y = F(x)$ певного типу, яка проходить через систему точок $M(x_i, y_i)$ ($i = 0, 1, 2, \dots, n$) (рисунок 6.1).

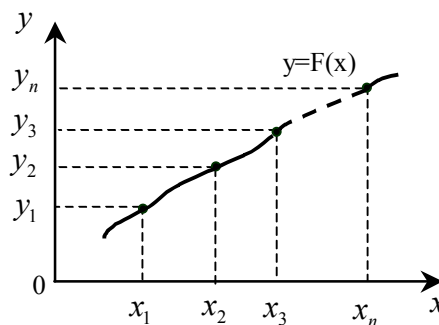


Рис. 6.1. Інтерполяція даних

В загальних випадках ця задача має нескінченну множину розв'язків чи зовсім не має розв'язку, але вона стає однозначною, якщо замість довільної функції $F(x)$ шукати поліном $P_n(x)$ ступеня не вище n , який задовольняє умову (6.1), тобто

$$P_n(x_0) = y_0, P_n(x_1) = y_1, \dots, P_n(x_i) = y_i, \dots, P_n(x_n) = y_n.$$

Інтерполяційну формулу $y = F(x)$, як правило, використовують для наближеного обчислення значень даної функції $f(x)$ для $x \neq x_i$ ($i = 0, 1, 2, \dots, n$). Така операція зветься інтерполяцією. Треба відзначити, що є інтерполяція в вузькому розумінні, коли $x \in [x_0, x_n]$, та екстрополіція, коли x знаходиться за межами інтервалу $[x_0, x_n]$, тобто $x < x_0$ чи $x > x_n$.

Розглянемо декілька методів інтерполяції.

6.1.1. Різницеві методи

Існує багато кінцево-різницевих методів інтерполяції. Найбільш поширеним є метод Ньютона для інтерполяції "уперед" (метод Ньютона-Грегорі). Інтерполяційний поліном в цьому випадку має вигляд:

$$P_n(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + \dots \\ \dots + C_n(x - x_0)(x - x_1) \dots (x - x_{n-1}).$$

Коефіцієнти C_i знаходять з рівнянь:

$$P_n(x_i) = y_i, \quad i = 0, 1, 2, \dots, n,$$

які дозволяють записати систему

$$\begin{aligned} C_0 &= y_0, \\ C_0 + C_1(x_1 - x_0) &= y_1, \\ C_0 + C_1(x_2 - x_0) + C_2(x_2 - x_0)(x_2 - x_1) &= y_2, \\ &\dots \\ C_0 + C_1(x_n - x_0) + \dots + C_n(x_n - x_0)(x_n - x_1)\dots(x_n - x_{n-1}) &= y_n. \end{aligned} \tag{6.2}$$

Це лінійна система рівнянь з трикутною матрицею.

Якщо прийняти крок $x_{i+1} - x_i = h$, то в області змінювання $x \in [x_0, x_n]$ отримаємо одновимірну рівномірну сітку та зможемо скористатись різницеvim зображенням системи (6.2), яке приводить до таких різницевих виразів для коефіцієнтів:

$$C_0 = y_0,$$

$$C_1 = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h},$$

Δy_0 – права різниця першого порядку в точці y_0 ;

$$C_2 = \frac{y_2 - 2y_1 + y_0}{2h^2} = \frac{\Delta^2 y_0}{2h^2},$$

$\Delta^2 y_0$ – права різниця другого порядку;

.....

$$C_j = \frac{\Delta^j y_0}{(j!)h^j},$$

$\Delta^j y_0$ – права різниця j -го порядку.

Тоді

$$P_n(x) = y_0 + \frac{\Delta y_0}{1!h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_1) + \dots$$

$$\dots + \frac{\Delta y_0}{n!h^n}(x - x_0)(x - x_1) \dots (x - x_n). \quad (6.3)$$

З практичної точки зору для визначення різниць вищих порядків використовують вираз:

$$\Delta^j y_i = \Delta(\Delta^{j-1} y_i) = \Delta^{j-1} y_{i+1} - \Delta^{j-1} y_i.$$

При $n = 1$ з (6.3) отримуємо формулу для лінійної інтерполяції

$$P_1(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0),$$

а якщо $n = 2$ – формулу параболічної чи квадратичної інтерполяції

$$P_2(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2h^2}(x - x_0)(x - x_1).$$

Якщо дана необмежена кількість значень функції x , то кількість n може бути будь-якою. Практично в цьому випадку n вибирають так, щоб різниця $\Delta^n y_i$ була постійною з заданою точністю. За початкове значення x_0 можна приймати будь-яке табличне значення аргументу x . Коли кількість значень функції скінченна, то кількість n обмежена та не може бути більше кількості значень функції y , зменшеної на одиницю.

Формула (6.3) носить назву першої інтерполяційної формули Ньютона. Цей вираз незручний для інтерполяції поблизу останніх значень y . В цьому випадку, як правило, використовують другу інтерполяційну формулу Ньютона, яка отримана при використанні лівих різниць від останнього значення (x_n, y_n) (інтерполяція “назад”). Тоді інтерполяційний поліном має вигляд:

$$\begin{aligned}
 P_n(x) = & C_0 + C_1(x - x_n) + C_2(x - x_n)(x - x_{n-1}) + \\
 & + C_3(x - x_n)(x - x_{n-1})(x - x_{n-2}) + \dots \\
 & \dots + C_n(x - x_n)(x - x_{n-1}) \dots (x - x_1).
 \end{aligned}$$

Коефіцієнти C_j визначаються таким чином:

$$C_0 = y_n$$

$$C_1 = \frac{\Delta y_{n-1}}{h} = \frac{\nabla y_n}{h},$$

∇y_n – ліва різниця першого порядку в точці y_n ,

$$C_2 = \frac{\Delta^2 y_{n-2}}{2!h^2} = \frac{\nabla^2 y_n}{2h^2},$$

$\nabla^2 y_n$ – ліва різниця другого порядку,

$$\begin{aligned}
 & \dots \dots \dots \\
 C_j = & \frac{\Delta^j y_{n-j}}{j!h^j} = \frac{\nabla^j y_n}{j!h^j},
 \end{aligned}$$

$\nabla^j y_n$ – ліва різниця j -го порядку.

Кінцевий вираз для другої інтерполяційної формули Ньютона:

$$\begin{aligned}
 P_n(x) = & y_n + \frac{\Delta y_{n-1}}{1!h}(x - x_n) + \frac{\Delta y_{n-2}^2}{2!h^2}(x - x_n)(x - x_{n-1}) + \dots \\
 & + \frac{\Delta^n y_0}{n!h^n}(x - x_n)(x - x_{n-1}) \dots (x - x_1).
 \end{aligned}$$

Інтерполяційні формули Ньютона можуть бути використані для екстраполяції функції. Якщо $x < x_0$, то зручно використовувати першу інтерполяційну формулу Ньютона, причому

$$\frac{x - x_0}{h} < 0.$$

Якщо $x > x_n$, то використовують другу інтерполяційну формулу Ньютона, де

$$\frac{x - x_n}{h} > 0.$$

Таким чином, перша інтерполяційна формула Ньютона, як правило, використовується для інтерполяції “уперед” та екстраполяції “назад”, а друга – для інтерполяції “назад” та екстраполяції “уперед”.

В формулах Ньютона використовують ліві та праві різниці. Використання центральних різниць для отримання інтерполяційних формул приводить до формул Гаусса, Стірлінга та Бесселя.

Треба відзначити, що центральні різниці використовуються не в звичайному вигляді, а шляхом застосування правих різниць при поступовому зсуві індексів вліво.

Ці формули зручно розглядати на $(2n + 1)$ рівновіддалених вузлах інтерполяції

$$x_{-n}, x_{-(n-1)}, \dots, x_{-1}, x_0, x_1, \dots, x_{n-1}, x_n,$$

причому

$$\Delta x_i = x_{i+1} - x_i = h \quad (i = -n, -(n-1), \dots, n-1),$$

а для функції $y = f(x)$ відомі її значення в цих вузлах $y_i = f(x_i)$.

Потрібно побудувати поліном $P(x)$ ступеня не вище $2n$ такий, що

$$P(x_i) = y_i.$$

Поліном $P(x)$ відшукується у вигляді:

$$\begin{aligned} P(x) = & C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + C_3(x - x_{-1}) \\ & (x - x_0)(x - x_1) + \dots + C_{2n-1}(x - x_{-(n-1)}) \dots \\ & \dots (x - x_{-1})(x - x_0)(x - x_1) \dots (x - x_{n-1})(x - x_n). \end{aligned} \quad (6.4)$$

Аналогічно інтерполяційним формулам Ньютона знаходимо

$$C_0 = y_0;$$

$$C_1 = \frac{\Delta y_0}{h};$$

$$C_2 = \frac{\Delta^2 y_{-1}}{2! h^2};$$

.....

$$C_{2n-1} = \frac{\Delta^{2n-1} y_{-(n-1)}}{(2n-1)! h^{2n-1}};$$

$$C_{2n} = \frac{\Delta^{2n} y_{-n}}{(2n)! h^{2n}}.$$

Підставляючи знайдені значення коефіцієнтів в (6.4), отримаємо першу інтерполяційну формулу Гаусса, яка містить різниці (таблиця 6.1):

$$\Delta y_0, \Delta^2 y_{-1}, \Delta^3 y_{-1}, \Delta^4 y_{-2}, \Delta^5 y_{-2}, \Delta^6 y_{-2}, \dots$$

Аналогічно можна отримати другу інтерполяційну формулу Гаусса, яка містить центральні різниці:

$$\Delta y_{-1}, \Delta^2 y_{-1}, \Delta^3 y_{-2}, \Delta^4 y_{-2}, \Delta^5 y_{-3}, \Delta^6 y_{-3}, \dots$$

Використавши середнє арифметичне першої та другої інтерполяційних формул Гаусса, отримаємо формулу Стірлінга. Ці формули дозволяють вивести інтерполяційну формулу Бесселя. Взагалі, використання інтерполяційних формул з центральними різницями доцільно в середині інтервалу, тоді як по його краях, як правило, використовують формули Ньютона. Застосування цих формул розглядається в таблиці 6.1.

Похибки інтерполяції для формул Ньютона можна оцінити відповідно для першої та другої формул як:

$$\Delta_n(x) = \frac{q(q-1)\dots(q-n)}{(n+1)!} \Delta^{n+1} y_0,$$

$$\Delta_n(x) = \frac{q(q+1)\dots(q+n)}{(n+1)!} \Delta^{n+1} y_n,$$

де $q = \frac{x - x_n}{h}$.

Для формули Стірлінга:

$$\Delta(x_n) = \frac{\Delta^{2n+1} y_{-(n-1)} + \Delta^{2n+1} y_{-n}}{2(2n+1)!} q(q^2-1)(q^2-2^2) \dots (q^2-n^2).$$

Таблиця 6.1

Застосування різницевих формул інтерполяції

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	Примітки
x_{-2}	y_{-2}		$\Delta^2 y_{-3}$		$\Delta^4 y_{-4}$	Друга Формула Ньютона
		Δy_{-2}		$\Delta^3 y_{-3}$		
x_{-1}	y_{-1}		$\Delta^2 y_{-2}$		$\Delta^4 y_{-3}$	
		Δy_{-1}		$\Delta^3 y_{-2}$		
x_0	y_0		$\Delta^2 y_{-1}$		$\Delta^4 y_{-2}$	Формула Стірлінга
		Δy_0		$\Delta^3 y_{-1}$		Формула Бесселя
x_1	y_1		$\Delta^2 y_0$		$\Delta^4 y_{-1}$	
		Δy_1		$\Delta^3 y_0$		
x_2	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$	
		Δy_2		$\Delta^3 y_1$		
x_3	y_3		$\Delta^2 y_2$		$\Delta^4 y_1$	Перша Формула Ньютона
		Δy_3		$\Delta^3 y_2$		

Для випадку нерівновіддалених значень аргументу можна отримати інтерполяційні формули, вживаючи визначення поділених різниць. Наприклад, відношення

$$[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

називається поділеною різницею першого порядку, а відношення

то записана система рівнянь буде задовільна.

Ця умова означає, що будь-який поліном $b_j(x)$ дорівнює нулю при кожному x_i , крім рівного x_j . Тому в загальному випадку поліном $b_j(x)$ має такий вигляд:

$$b_j(x) = C_j(x - x_0)(x - x_1)\dots(x - x_{j-1})(x - x_{j+1})\dots(x - x_n).$$

Якщо $b_j(x_j) = 1$, то коефіцієнти C_j визначаються з виразу:

$$C_0 = 1/(x_j - x_0)\dots(x_j - x_{j-1})(x_j - x_{j+1})\dots(x_j - x_n).$$

Для полінома, який шукаємо, отримаємо:

$$P_n(x) = \sum_{j=0}^n y_j * \frac{(x - x_0)(x - x_1)\dots(x - x_{j-1})(x - x_{j+1})\dots(x - x_n)}{(x_j - x_0)(x_j - x_1)\dots(x_j - x_{j-1})(x_j - x_{j+1})\dots(x_j - x_n)}.$$

Вводячи позначення

$$L_j(x) = (x - x_0)(x - x_1)\dots(x - x_{j-1})(x - x_{j+1})\dots(x - x_n),$$

записуємо

$$P_n(x) = \sum_{j=0}^n y_j \frac{L_j(x)}{L_j(x_j)}.$$

Треба відзначити дві головні властивості поліномів Лагранжа:

$$1) \sum_{j=0}^n \frac{L_j(x)}{L_j(x_j)} = 1;$$

2) якщо $P_n(x)$ лінійно залежить від y_j , то слушний принцип суперпозиції: інтерполяційний поліном суми декількох функцій дорівнює сумі інтерполяційних поліномів доданків.

Похибка при інтерполяції за Лагранжем може бути оцінена таким чином:

$$|\Delta(x_n)| \leq \frac{M_{n+1}}{(n+1)!} (x - x_0)(x - x_1)\dots(x - x_n),$$

$$\text{де } M_{n+1} = \max_{x_0 \leq x \leq x_n} |f^{(n+1)}(x)|.$$

6.1.3. Сплайн-інтерполяція

Сплайни порівняно недавно почали широко вживатися в обчислювальній математиці. В машинобудівельному кресленні вони застосовуються багато років, тому що це і є лекала чи гнучкі лінійки, які деформуються так, щоб з їх допомогою можна було провести криву через задані точки (x_i, y_i) .

Можна показати (вживаючи теорію згину бруса при малих деформаціях), що сплайн – це група сполучених кубічних багаточленів, в місцях сполучення яких перша та друга похідні безперервні. Такі функції звуться кубічними сплайнами. Для їх побудування необхідно задати коефіцієнти, які однозначно визначають поліном у проміжку між двома точками.

Наприклад, у випадку, який показаний на рисунку 6.2, необхідно задати всі кубічні функції $q_1(x), q_2(x), \dots, q_m(x)$. В найбільш загальному випадку ці багаточлени мають такий вигляд:

$$q_i(x) = k_{1i} + k_{2i}x + k_{3i}x^2 + k_{4i}x^3, \quad i=1, 2, \dots, m$$

де k_{ji} – постійні, які визначені вказаними умовами ($j=1, 2, 3, 4$).

Перші $(2m)$ умов потребують, щоб сплайни стикалися в заданих точках:

$$\begin{aligned} q_i(x_i) &= y_i, & i=1, 2, \dots, m, \\ q_{i+1}(x_i) &= y_i, & i=0, 1, \dots, m-1. \end{aligned}$$

Наступні $(2m-2)$ умов потребують, щоб в місцях дотику сплайнів були рівні перші та другі похідні

$$\begin{aligned} q'_{i+1}(x_i) &= q'_i(x_i), & i=1, \dots, m-1, \\ q''_{i+1}(x_i) &= q''_i(x_i), & i=1, \dots, m-1. \end{aligned}$$

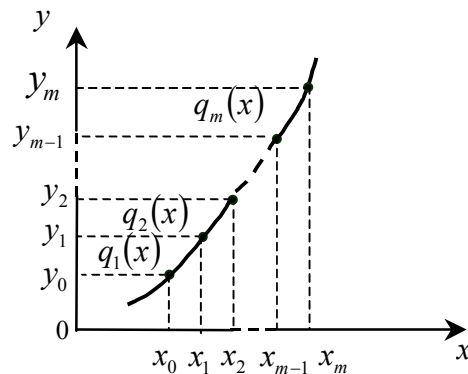


Рис. 6.2. Сплайн-інтерполяція

Система алгебраїчних рівнянь має розв'язок, якщо кількість рівнянь дорівнює кількості невідомих. Для цього необхідні ще два рівняння. Як правило, використовують такі додаткові умови:

$$q_1'(x_0) = 0; \quad q_m''(x_m) = 0.$$

Отриманий таким чином сплайн зветься “природним кубічним сплайном”. При знайдених коефіцієнтах сплайна використовують цю кусково-гладку поліноміальну функцію для інтерполяції.

Якщо спеціально вибрати вигляд кубічних багаточленів, можна значно спростити задачу (зменшити кількість рівнянь). В випадку, коли окремі кубічні рівняння мають вигляд:

$$q_i(x) = ty_i + \bar{t}y_{i-1} + \Delta x_i[(k_{i-1} - d_i)t^2\bar{t} - (k_i - d_i)t^2\bar{t}], \quad i=1, 2, \dots, m,$$

$$\text{де } \Delta x_i = x_i - x_{i-1}, \quad t = \frac{x - x_{i-1}}{\Delta x_i}, \quad \bar{t} = 1 - t,$$

$$\text{а } \Delta y_i = y_i - y_{i-1}, \quad \frac{\Delta y_i}{\Delta x_i} = d_i,$$

кожне з рівнянь $q_i(x)$ містить тільки два невідомих коефіцієнти. Після того, як перше рівняння записано, з кожним наступним рівнянням додається тільки один невідомий коефіцієнт. При цьому при $x = x_{i-1}$, $t = 0$, $\bar{t} = 1$, а при $x = x_i$, $t = 1$, $\bar{t} = 0$,

Отже, всі умови, крім умов для других похідних, задовольняються. Другі похідні виражені для внутрішніх точок відношеннями:

$$k_{i-1}\Delta x_{i+1} + 2k_i(\Delta x_i + \Delta x_{i+1}) + k_{i+1}\Delta x_i = 3(d_i\Delta x_{i+1} + d_{i+1}\Delta x_i),$$

а для двох зовнішніх:

$$2k_0 + k_1 = 3d_1 \quad \text{та} \quad k_{m-1} + 2k_m = 3d_m.$$

Таким чином, система рівнянь, яку розв'язуємо, є лінійною, а її матриця—тридіагональною:

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ \Delta x_2 & 2(\Delta x_1 + \Delta x_2) & \Delta x_1 & \Delta x_2 \\ 0 & \Delta x_3 & 2(\Delta x_2 + \Delta x_3) & 2(\Delta x_{m-1} + \Delta x_m) & \Delta x_{m-1} \\ & & \Delta x_m & 1 & 2 \end{bmatrix}^*$$

$$* \begin{bmatrix} k_0 \\ k_1 \\ \cdot \\ \cdot \\ \cdot \\ k_m \end{bmatrix} = 3 \begin{bmatrix} d_1 \\ d_1 \Delta x_2 + d_2 \Delta x_1 \\ d_2 \Delta x_3 + d_3 \Delta x_2 \\ \dots \\ d_{m-1} \Delta x_m + d_m \Delta x_{m-1} \\ d_m \end{bmatrix}.$$

Методи розв'язування таких систем добре розроблені.

В багатьох випадках метод сплайнів є найбільш зручним, тому що це дозволяє отримати аналітичну кусково-поліноміальну функцію. Існують сплайни більш вищих порядків. Вживання цього методу можливо і в інших галузях обчислювальної математики, наприклад, в чисельному інтегруванні і розв'язанні диференціальних рівнянь.

6.2. Апроксимація даних

Апроксимація взагалі – це наближений опис однією функцією (апроксимувальною) заданого вигляду іншої функції (апроксимовної), яка задається у будь-якому вигляді (при апроксимації даних вона задається у вигляді масивів даних).

Існує два головних підходи до апроксимації даних. При одному з них вимагають, щоб апроксимувальна крива (можливо кусково-гладка) проходила через всі точки, які задані таблицею. Це можна зробити з допомогою методів інтерполяції, які були розглянуті в попередньому розділі. При іншому підході дані апроксимують простою функцією, яка використовується при всіх табличних значеннях, але не обов'язково, щоб вона проходила через всі точки. Такий підхід зветься припасуванням кривої, яку прагнуть провести так, щоб її відхилення від табличних даних був мінімальним. Як правило, користуються методом найменших квадратів, тобто зводять до мінімуму суму квадратів різниць між значенням функції, яка визначена обраною кривою, та таблицею.

Нехай у таблиці задана $(n + 1)$ точка $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ і треба знайти апроксимувальну криву $g(x)$ в діапазоні $x_0 \leq x \leq x_n$ (рисунок 6.3). В цьому випадку похибка в кожній табличній точці буде

$$\varepsilon_i = g(x_i) - y_i.$$

Тоді сума квадратів похибок визначається виразом:

$$E = \sum_{i=0}^n [g(x_i) - y_i]^2.$$

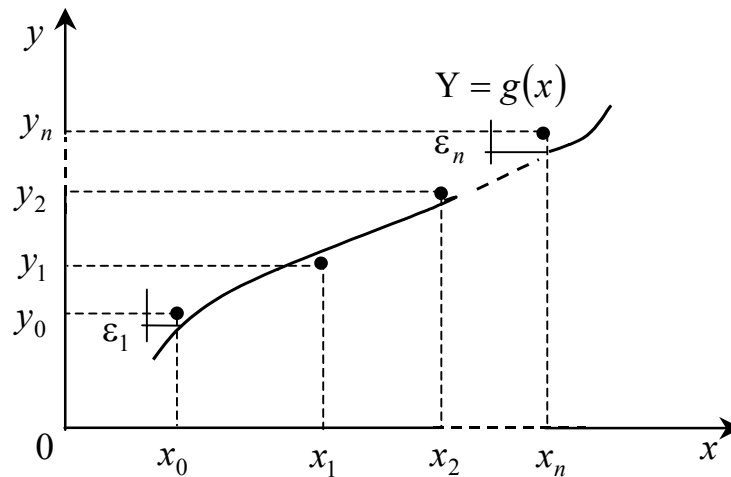


Рис. 6.3. Апроксимація даних

Як правило, функцію $g(x)$ обирають у вигляді лінійної комбінації вибраних функцій $g_k(x)$

$$g(x) = C_1 g_1(x) + C_2 g_2(x) + \dots + C_k g_k(x).$$

Умова мінімуму E визначається рівнянням:

$$\frac{\partial E}{\partial C_1} = \frac{\partial E}{\partial C_2} = \dots = \frac{\partial E}{\partial C_k} = 0.$$

Відомо, що

$$E = \sum_{i=0}^n [C_1 g_1(x_i) + C_2 g_2(x_i) + \dots + C_k g_k(x_i) - y_i^2]$$

ця умова еквівалентна системі рівнянь:

$$\frac{\partial E}{\partial C_1} = 2 \sum [C_1 g_1(x_1) + \dots + C_k g_k(x_i) - y_i] g_1(x_1) = 0;$$

.....

$$\frac{\partial E}{\partial C_k} = 2 \sum [C_1 g_1(x_i) + \dots + C_k g_k(x_i) - y_i] g_k(x_i) = 0.$$

Цю систему можна записати у матричній формі:

$$\begin{bmatrix} \sum g_1^2(x_i) & \sum g_1(x_i)g_2(x_i) & \dots & \sum g_1(x_i)g_k(x_i) \\ \dots & \dots & \dots & \dots \\ \sum g_1(x_i)g_k(x_i) & \dots & \dots & \sum g_k^2(x_i) \end{bmatrix}^* \quad (6.5)$$

$$* \begin{bmatrix} C_1 \\ C_2 \\ \cdot \\ C_k \end{bmatrix} = \begin{bmatrix} \sum g_1(x_i)y_i \\ \dots \\ \sum g_k(x_i)y_i \end{bmatrix}.$$

Елементи матриці в лівій частині та вектора-стовпця в правій визначаються табличними даними, тому отримана система k лінійних рівнянь з k невідомими може бути розв'язана. Вибір функції $g(x)$ повинен здійснюватися з урахуванням характеру табличних даних (періодичності, властивості симетрії, існування асимптотики та т. п.).

Іноді таблицю розбивають на декілька частин та добирають окрему апроксимувальну криву для кожної частини. Такий підхід задовольняє ті випадки, коли дані відповідають різним фізичним станам системи.

Залишкова середня квадратична похибка апроксимації оцінюється:

$$\Delta = \sqrt{E/(n+1)}.$$

Якщо при побудові апроксимувальної функції використовуються ортогональні поліноми, для яких

$$\sum g_j(x_i)g_k(x_i) = 0, \text{ якщо } j \neq k,$$

то система (6.5) спрощується, і матриця стає діагональною. Коефіцієнти визначаються зі співвідношень

$$C_j = \sum_{i=0}^n g_j(x_i)y_i / \sum_{i=0}^n g_j^2(x_i).$$

Це спрощує задачу, і тому в багатьох стандартних програмах припасування кривих використовують ортогональні поліноми.

6.3. Статистична обробка даних

При обробці результатів експериментів в вимірювальній техніці (наприклад, для оцінки похибки вимірювань), автоматичі (розв'язування задач ідентифікації, оптимального управління), статистичній радіотехніці виникає необхідність оцінки характеристик випадкової величини.

В якості оцінки \bar{x} невідомого математичного сподівання m_X випадкової величини X використовується середнє арифметичне результатів n незалежних випробувань

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n},$$

а для оцінки дисперсії σ_X^2 – співвідношення

$$\delta^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}.$$

При припущенні нормального закону розподілу величини x можна показати, що величина

$$T = \frac{\bar{x} - m_X}{\delta / \sqrt{n}}$$

має t -розподіл Стюдента з $k = n - 1$ ступенями вільності. Звідси можна визначити надійний інтервал для справжнього значення x : за відомими значеннями надійної імовірності P з таблиці 6.2 знаходимо ε , звідки

$$\Delta = \varepsilon * \frac{\delta}{\sqrt{n}}.$$

Таким чином, випадкова величина x розподілена за нормальним законом з математичним сподіванням m_X і дисперсією σ_X^2 . Справжнє значення x знаходиться в інтервалі $(m_X - \Delta, m_X + \Delta)$ з надійною імовірністю P .

Для оцінки виду закону розподілу найбільше застосування мають критерії Колмогорова і Пірсона, які дозволяють на основі порівняння емпіричної функції розподілу $f_x^*(x)$, одержаної в вигляді гістограми в результаті обробки експериментальних даних, з гіпотетичною $f_x(x)$, яка

відповідає запропонованій гіпотезі, зробити висновки про їх збігання чи незбігання при рівні значущості α , який визначається як імовірність того, що буде відхилена вірна гіпотеза.

В критерії Колмогорова мірою є величина

$$\lambda = \left| f_X(x) - f_X^*(x) \right|_{\max} \sqrt{n},$$

яку порівнюють з критичним значенням, заданим з таблиці 6.3.

Таблиця 6.2

Значення для надійного інтервалу $-\varepsilon < t < \varepsilon$, де величина t має розподіл Стюдента в залежності від надійної імовірності p і числа ступенів вільності k

k	$p=0,90$	$p=0,95$	$p=0,99$
1	6,310	12,71	63,7
2	2,920	4,30	9,92
3	2,350	3,18	5,84
4	2,130	2,77	4,60
5	2,020	2,57	4,03
6	1,943	2,45	3,71
7	1,895	2,36	3,50
8	1,860	2,31	3,36
9	1,833	2,26	3,25
10	1,812	2,23	3,17
11	1,796	2,20	3,11
12	1,782	2,18	3,06
13	1,771	2,16	3,01
14	1,761	2,14	1,98
15	1,753	2,13	2,95
16	1,746	2,12	2,92
17	1,740	2,11	2,90
18	1,734	2,10	2,86
19	1,729	2,09	2,86
20	1,725	2,08	2,84
22	1,717	2,07	2,82
24	1,711	2,06	2,80
26	1,706	2,06	2,78
28	1,701	2,05	2,76
30	1,697	2,04	2,75
40	1,684	2,02	2,70
60	1,671	2,00	2,66
120	1,658	1,98	2,62
	1,645	1,96	2,58

Таблиця 6.3

Критичні значення λ_0 в залежності від рівня значущості

α	0,50	0,40	0,30	0,20	0,10	0,05	0,02	0,001	0,001
λ_0	0,828	0,895	0,974	1,073	1,224	1,358	1,520	1,627	1,950

При $\lambda < \lambda_{\text{гп}}$ гіпотеза про збігання $f_X(x)$ и $f_X^*(x)$ приймається.

В критерії Пірсона обчислюється величина

$$\chi^2 = \sum_{i=0}^k \frac{[f_X(x_i) - f_X^*(x_i)]^2}{f_X(x_i)},$$

де k – число розрядів гістограми (дискретних значень $f_X(x_i)$).

З таблиці 6.4 визначають критичне значення χ^2 , зважаючи на α і число ступенів вільності

$$r = k - l - 1,$$

де l – число параметрів, що їх містить в собі закон розподілу (для нормального $l=2$, пуассонівського $l=1$ і т.д.).

При $\chi^2 < \chi_{\text{гп}}^2$ гіпотеза приймається.

Якщо порівнюють аналітично одержані закони розподілу імовірностей, то мірою їх близькості служить значення середньої квадратичної похибки.

Для оцінки взаємозалежності випадкових величин, між якими існує стохастичний зв'язок, використовується коефіцієнт кореляції (n – об'єм вибірки)

$$R_{xy} = \frac{1}{n-1} \frac{\sum_{i=1}^n (x_i - m_x)(y_i - m_y)}{\sigma_x \sigma_y}.$$

При визначенні взаємозалежності значень випадкових величин в різні моменти часу коефіцієнт кореляції оцінюється за формулою

$$R_X(\tau) = \frac{1}{n-m-1} \frac{\sum_{i=1}^{n-m} [x(t_i) - m_X][x(t_i + \tau) - m_X]}{D_X},$$

Таблиця 6.4

Критичні точки розподілу

x – випадкова величина, яка розподілена за законом χ^2 з ступенями вільності k (таблиця містить значення α , що одержані із умови $P(x) \leq \alpha$)

Число ступенів вільності k	$\alpha = 0,01$	$\alpha = 0,025$	$\alpha = 0,05$	$\alpha = 0,95$	$\alpha = 0,975$	$\alpha = 0,99$
1	6,6	6,0	3,8	0,0039	0,00098	0,00016
2	9,2	7,4	6,0	0,103	0,051	0,020
3	11,3	9,4	7,8	0,352	0,216	0,115
4	13,3	11,1	9,5	0,711	0,484	0,297
5	15,1	12,8	11,1	1,15	0,831	0,554
6	16,8	14,4	12,6	1,64	1,24	0,872
7	18,5	16,0	14,1	2,17	1,69	1,24
8	20,1	17,5	15,5	2,73	2,18	1,65
9	21,7	19,0	16,9	3,33	2,70	2,09
10	23,2	20,5	18,3	3,94	3,25	2,56
11	24,7	21,9	19,7	4,57	3,82	3,05
12	26,2	23,3	21,0	5,23	4,40	3,57
13	27,7	24,7	22,4	5,89	5,01	4,11
14	29,1	26,1	23,7	6,57	5,63	4,66
15	30,6	27,5	25,0	7,26	6,26	5,23
16	32,0	28,8	26,3	7,96	6,91	5,81

де $x(t_i)$ – значення випадкової величини X в момент часу t_i , а $x(t_i + \tau)$ – в момент часу, який відрізняється від t_i на інтервал τ . Таким чином, $x(t_i) = x_i$, $x(t_i + \tau) = x_j$, τ – інтервал часу між i та j значеннями x , $i - j = m$.

Інтервал кореляції визначається як відрізок часу, за який кореляційна функція зменшується на 95 %.

Обчислення кореляційної функції за відомими масивами даних x та y за допомогою наведених формул не викликає труднощів, а апроксимація виду кореляційної функції типовими кореляційними функціями (таблиця 6.5) може здійснюватися за методом найменших квадратів.

Таблиця 6.5

Типові кореляційні функції

Вигляд	Параметри
$R_x(\tau) = \sigma_x^2 (1 - \alpha \tau),$ $\tau < 1/\alpha$	$\alpha = (\sigma_x^2 - R_x(\tau^*)) / \sigma_x^2 \tau^*,$ $R_x(\tau^*)$ – відоме значення кореляційної функції
$R_x(\tau) = \sigma_x^2 e^{-\alpha \tau }$	$\alpha = \frac{1}{\tau^*} \ln \frac{\sigma_x^2}{R_x(\tau^*)}$
$R_x(\tau) = \sigma_x^2 e^{-\alpha^2 \tau^2}$	$\alpha = \frac{1}{\tau^*} \sqrt{\ln \frac{\sigma_x^2}{R_x(\tau^*)}}$
$R_x(\tau) = \sigma_x^2 e^{-\alpha \tau } (1 + \alpha \tau)$	$\alpha \approx 4,5/\tau_k^{max}$
$R_x(\tau) = \sigma_x^2 e^{-\alpha \tau } \cos \beta\tau$	$\alpha = \frac{1}{\tau_2^*} \ln \frac{\sigma_x^2 \cos \frac{\pi\tau_2^*}{2\tau_1^*}}{R_x(\tau_2^*)},$ $\beta = \pi / 2\tau_1^*$ при двох відомих значеннях кореляційної функції $R_x(\tau)$, причому $R_x(\tau_1^*) = 0$.

6.4. Чисельне інтегрування

В багатьох задачах, що пов'язані з аналізом, ідентифікацією, оцінкою якості різних засобів та систем автоматики та управління, виникає необхідність обчислення певних інтегралів.

Якщо функція $f(x)$ неперервна на відрізку $[a, b]$ й відома її первісна $F(x)$, то визначений інтеграл від a до b може бути обчислений за формулою Ньютона-Лейбніца

$$I = \int_a^b f(x)dx = F(b) - F(a),$$

де $F'(x) = f(x)$.

Але часто точно обчислити інтеграл важко через велику складність аналітичних перетворень, а інколи це взагалі неможливо (в випадках невластних інтегралів), чи коли підінтегральна функція задана набором числових даних, наприклад, отриманих з експерименту.

Задача чисельного інтегрування функції полягає в обчисленні значення визначеного інтегралу на основі ряду значень підінтегральної функції. Графічно інтеграл визначається площею, що обмежується графіком функції

$$y = f(x).$$

Найбільш відомими методами знаходження визначених інтегралів є:

- методи Ньютона-Котеса, Гаусса, Чебишева, які основані на використанні так званих квадратурних формул, отриманих заміною $f(x)$ інтерполяційними багаточленами;
- методи Монте-Карло, основані на використанні статистичних моделей.

6.4.1. Формули Ньютона-Котеса

Для виведення формул Ньютона-Котеса інтеграл зображується у вигляді

$$\int_a^b f(x)dx = \sum_{i=0}^n A_i f(x_i) + \Delta, \quad (6.6)$$

де x_i – вузли інтерполяції; A_i – коефіцієнти, які залежать від вигляду формули; Δ – похибка квадратурної формули.

Замінюючи в (6.6) підінтегральну функцію відповідним інтерполяційним поліномом Лагранжа для n рівновіддалених вузлів з кроком $h = \frac{b-a}{n}$, можна одержати наступну формулу для розрахунку коефіцієнтів A_i при довільній кількості вузлів.

$$A_i = \frac{b-a}{n} \frac{(-1)^{n-1}}{i!(n-1)!} \int_0^n \frac{q(q-1)\dots(q-n)}{(q-i)} dq, \quad (6.7)$$

де $q = \frac{x-a}{h}$ – приведена змінна.

Звичайно коефіцієнти $H_i = \frac{A_i}{b-a}$ називають коефіцієнтами Котеса.

При цьому формула (6.6) приймає вигляд

$$\int_a^b f(x) dx = (b-a) \sum_{i=0}^n H_i f(x_i) \quad (6.8)$$

і має такі властивості:

$$\sum_{i=0}^n H_i = 1 \quad i \quad H_i = H_{n-1}.$$

При $n=1$ і $n=2$ із (6.7) та (6.8) отримаємо формули трапецій і Сімпсона:

$$I = \frac{h}{2} [f(x_0) + f(x_1)],$$

$$I = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)].$$

В таблиці 6.6 наведені значення коефіцієнтів для $n=1,2,\dots,8$.

Похибки формул трапецій і Сімпсона визначаються, відповідно, із виразів

$$\Delta = -\frac{h^3}{12} M_2 \quad \text{та} \quad \Delta = -\frac{h^5}{90} M_4,$$

де M_2 і M_4 – максимальні значення другої та четвертої похідної $f(x)$ при $x \in (a, b)$.

Складові формули Ньютона-Котеса отримаємо шляхом комбінації простих формул. Наприклад, для формул трапецій і Сімпсона (для парних n):

$$I = \frac{h}{2} [f(x_0) + 2f(x_1) + 2f(x_1) + \dots + f(x_n)],$$

$$I = \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \dots + f(x_n)].$$

Причому похибки складових формул будуть, відповідно,

$$\Delta = -n \frac{h^3}{12} M_2 \quad \text{та} \quad \Delta = -n \frac{h^5}{180} M_4.$$

Аналогічно можна отримати складові формули Ньютона – Котеса більш високих порядків.

Для оцінки похибки на практиці можна користуватись методами Рунге (екстраполяції Річардсона), що розглянуті в главі 4.

Таблиця 6.6

Коефіцієнти Котеса

$\hat{H}_i = H_i$ N	\hat{H}_0	\hat{H}_1	\hat{H}_2	\hat{H}_3	\hat{H}_4	\hat{H}_5	\hat{H}_6	\hat{H}_7	\hat{H}_8	Загальний знаменник N
1	1	1								2
2	1	4	1							6
3	1	3	3	1						8
4	7	32	12	32	7					90
5	19	75	50	50	75	19				288
6	41	216	27	272	27	216	41			840
7	751	3577	1223	2989	2989	1323	3577	751		17280
8	989	5888	-928	10496	-4540	10496	-928	5888	989	28350

6.4.2. Формула Чебишева

Формула (6.6) може бути приведена до вигляду

$$\int_{-1}^1 f(t) dt = \sum_{i=1}^n A_i f(t_i) \quad (6.9)$$

заміною змінних

$$x = \frac{a+b}{2} + \frac{b-a}{2} t.$$

При виведенні формули Чебишева використовуються такі умови:

- коефіцієнти A_i рівні між собою;
- квадратурна формула (6.9) точна для всіх поліномів до степеня n включно.

При цих умовах формула (6.9) має вигляд:

$$\int_{-1}^1 f(t) dt = \frac{2}{n} \sum_{i=1}^n f(t_i). \quad (6.10)$$

Для знаходження t_i використовуємо другу умову, згідно з якою формула (6.10) повинна бути точною для функції вигляду

$$f(t) = t^k, \quad k = 1, \dots, n.$$

Після підстановки цих функцій в (6.10) отримаємо систему рівнянь

$$\begin{cases} t_1 + t_2 + \dots + t_n = 0; \\ t_1^2 + t_2^2 + \dots + t_n^2 = \frac{n}{3}; \\ t_1^n + t_2^n + \dots + t_n^n = \frac{n[1 - (-1)^{n+1}]}{2(n+1)}. \end{cases} \quad (6.11)$$

Система рівнянь (6.11) має розв'язок при $n < 8$ та $n = 9$. В цій обмеженій точності і полягає недолік формули Чебишева. Значення t_i для різних n наведені в таблиці 6.7.

Для довільного інтервалу (a, b) формула (6.10) приймає вигляд

$$I = \frac{b-a}{n} \sum_{i=1}^n f(x_i),$$

де

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} t_i.$$

Похибка обчислень за методом Чебишева:

$$\Delta = \int_a^b \frac{\left(x - \frac{a+b}{2}\right)^{n+1}}{(n+1)!} f^{(n+1)}(x) dx - \frac{b-a}{n(n+1)!} \sum_{i=1}^n \left(x_i - \frac{a+b}{2}\right)^{n+1} f^{(n+1)}(x).$$

Таблиця 6.7

Значення абсцис t_i в формулі Чебишева

n	i	t_i	n	i	t_i
2	1;2	$\mp 0,577350$	6	1;6	$\mp 0,866247$
3	1;3	$\mp 0,707107$		2;5	$\mp 0,422519$
	2	0		3;4	$\mp 0,266635$
4	1;4	$\mp 0,794654$	7	1;7	$\mp 0,883862$
	2;3	$\mp 0,187592$		2;6	$\mp 0,529657$
5	1;5	$\mp 0,832498$		3;5	$\mp 0,323912$
	2;4	$\mp 0,374513$	4	0	
	3	0			

6.4.3. Формула Гаусса

Формула Гаусса називається формулою найвищої алгебраїчної точності. Для формули вигляду (6.9) найвища точність може бути досягнута для поліномів степеня $(2n - 1)$, які визначаються $2n$ постійними t_i і A_i ($i = 1, 2, \dots, n$).

Завдання полягає у визначенні коефіцієнтів A_i і абсцис точок t_i .

Для знаходження цих постійних розглянемо виконання формули (6.9) для функцій вигляду

$$f(t) = t^k, k = 0, 1, \dots, 2n - 1.$$

Враховуючи, що

$$\int_{-1}^1 t^k dt = \begin{cases} 2/(k+1), \\ 0 \end{cases},$$

отримаємо систему рівнянь

$$\begin{cases} \sum_{i=1}^n A_i = 2; \\ \sum_{i=1}^n A_i t_i = 0; \\ \sum_{i=1}^n A_i t_i^2 = 1; \\ \sum_{i=1}^n A_i t_i^{2n-2} = \frac{2}{2n-1}; \\ \sum_{i=1}^n A_i t_i^{2n-1} = 0. \end{cases} \quad (6.12)$$

Ця система нелінійна, і її звичайне розв'язання пов'язане із значними обчислювальними труднощами. Але якщо використовувати систему для поліномів вигляду

$$f(t) = t^k P_n(t), \quad k = 0, 1, \dots, n-1,$$

де $P_n(t)$ – поліном Лежандра, тоді її можна звести до лінійної відносно коефіцієнтів A_i з заданими точками t_i . Оскільки степені поліномів в співвідношенні не перевищують $2n-1$, повинна виконуватися система (6.12) і формула (6.9) приймає вигляд

$$\int_{-1}^1 t^k P_n(t) dt = \sum_{i=1}^n A_i t_i^k P_n(t_i). \quad (6.13)$$

В результаті властивості ортогональності ліва частина виразу (6.13) дорівнює 0, тоді

$$\sum_{i=1}^n A_i t_i^k P_n(t_i) = 0,$$

що завжди забезпечується при будь-яких значеннях A_i в точках t_i , які відповідають кореням відповідних поліномів Лежандра.

Підставляючи ці значення t_i в систему (6.12) і враховуючи перші n рівнянь, можна визначити коефіцієнти A_i .

Формула (6.9), де t_i – нулі полінома Лежандра $P_n(t)$, а $A_i, i = 1, 2, \dots, n$ визначаються із системи (6.12), називається формулою Гаусса.

Значення t_i, A_i для різних n наведені в таблиці 6.8.

Для довільного інтервалу (a,b) формула для методу Гаусса приймає вигляд

$$I = \frac{b-a}{2} \sum_{i=1}^n A_i f(x_i),$$

де

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} t_i.$$

Оцінка похибки формули Гаусса з n вузлами визначається із співвідношення

$$\Delta \leq \frac{(b-a)^{2n+1} (n!)^4 M_{2n}}{2^{2n+1} [(2n)!]^3 (2n+1)},$$

де M_{2n} – максимальне значення $2n$ похідної на ділянці $[a,b]$.

Таблиця 6.8

Елементи формули Гаусса

n	i	t_i	A_i
1	1	0	2
2	1;2	$\mp 0,657735027$	1
3	1;3	$\mp 0,677459667$	$\frac{5}{9}=0,55555556$
	2	0	$\frac{8}{9}=0,88888889$
4	1;4	$\mp 0,86113631$	0,34785484
	2;3	$\mp 0,33998104$	0,65214516
6	1;6	$\mp 0,93246951$	0,17132450
	2;5	$\mp 0,66120939$	0,36076158
	3;4	$\mp 0,238619119$	0,46791394
7	1;7	$\mp 0,94910791$	0,12948496
	2;6	$\mp 0,74153119$	0,27970540
	3;5	$\mp 0,40584515$	0,38183006
	4	0	0,41795918
8	1;8	$\mp 0,96028986$	0,10122854
	2;7	$\mp 0,79666648$	0,22238104
	3;6	$\mp 0,52553142$	0,31370664
	4;5	$\mp 0,18343464$	0,36268378

6.4.4. Алгоритми застосування чисельних методів

Послідовність застосування формул Ньютона-Котеса.

1. Вибір формули і знаходження (за допомогою таблиці 6.6) коефіцієнтів A_i .
2. Складання алгоритму та програми, причому:
 - у випадку задання дискретних значень $y_i = f(x_i)$ через крок h ці значення підставляються в (6.6);
 - у випадку задання функції $y = f(x)$ значення $y_i = f(x_i)$ вираховується, причому $x_i = x_0 + ih = a + ih (a \leq x \leq b)$.
3. Оцінка похибки.

Послідовність застосування методу Гаусса:

1. Вибір порядку методу і знаходження (за допомогою таблиці 6.8) коефіцієнтів A_i і значень $t_i (-1 \leq t \leq 1)$.
2. Розбиття інтервалу $a \leq x \leq b$ на l інтервалів (рисунок 6.4).
3. Знаходження значень інтервалу для кожного інтервалу ($j = 1, \dots, l$)

$$I = \sum_{j=1}^l I_j.$$

При цьому значення абсцис x_i всередині кожного інтервалу j знаходяться за формулою

$$x_i = \frac{a_{j+1} + a_j}{2} + \frac{a_{j+1} - a_j}{2} t_i, \quad (6.14)$$

де

$$a_{j+1} = a_j + h; \quad a_1 = a;$$

$$a_{l+1} = b; \quad j = 1, \dots, l;$$

$$h = \frac{b - a}{n}.$$

I_j знаходяться за формулою:

$$I_j = \int_{a_j}^{a_{j+1}} f(x) dx = \frac{(a_{j+1} - a_j)}{2} \sum_{i=1}^n A_i f(x_i). \quad (6.15)$$

4. Оцінка похибки.

В методі Чебишева послідовність дій аналогічна методу Гаусса, але в пункті 1 коефіцієнти t_i беруться з таблиці 6.7, а в пункті 3 для знаходження j -го інтегралу використовується формула

$$I_j = \int_{a_j}^{a_{j+1}} f(x) dx = \frac{(a_{j+1} - a_j)}{n} \sum_{i=1}^n f(x_i), \quad (6.16)$$

де x_i оцінюється тим же чином, що і в методі Гаусса, за формулою (6.14)

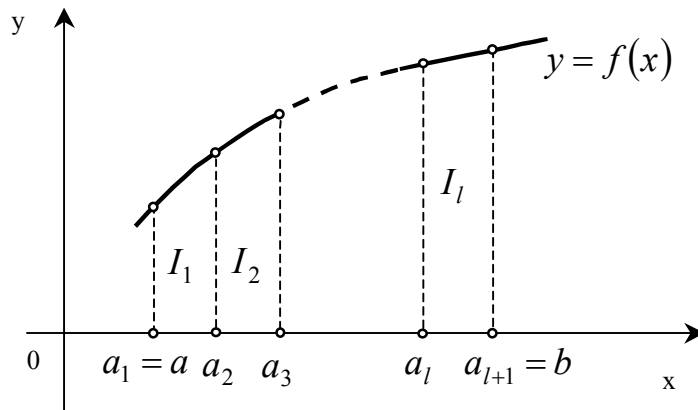


Рис. 6.4.

6.4.5 Метод Монте-Карло

Метод чисельного інтегрування Монте-Карло – це найбільш відоме застосування статичного моделювання для розв'язання прикладних математичних задач.

Якщо з послідовністю випадкових чисел $\{x_i\} \in X$ з законом розподілу ймовірностей $f_x(x)$ провести функціональне перетворення

$$y_i = \varphi(x_i),$$

то математичне очікування отриманої послідовності випадкових чисел $\{y_i\} \in Y$

$$m_Y = \int_{-\infty}^{\infty} \varphi(x) f_x(x) dx \quad (6.17)$$

при обсязі вибірки більше декількох тисяч чисел з достатньо високою точністю може бути оцінено за формулою

$$m_Y = \frac{1}{n} \sum_{i=1}^n y_i \quad (6.18)$$

Введемо в вирази (6.14), (6.15) так звану функцію індикатора області

$$1[a, b, x] = \begin{cases} 1, & a \leq x \leq b; \\ 0, & x < a, x > b. \end{cases}$$

Якщо тепер обрати функцію

$$\varphi(x) = \frac{f(x)}{f_x(x)},$$

то кінцевий вираз буде мати вигляд

$$I = m_Y = \int_a^b f(x) dx = \frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{f_x(x_i)} 1[a, b, x_i].$$

Алгоритм обчислення визначеного інтегралу за методом Монте-Карло наведено на рисунку 6.5.

Похибка методу Монте-Карло визначається похибкою генерації псевдовипадкової послідовності чисел, що згенеровані на ЕОМ, та обсягом вибірки. Вона може бути оцінена із співвідношення

$$\Delta = \frac{1}{2\sqrt{n(1-P)}}, \quad (6.19)$$

де P – гарантована ймовірність влучання похибки в інтервал $[-\Delta; +\Delta]$.

Кількість випробувань n не залежить від кратності інтегралу, тому метод Монте-Карло знаходить застосування для обчислення багатократних інтегралів, де застосовувати інші методи чисельного інтегрування неефективно через сильне збільшення кількості обчислювальних операцій.

Розглянемо послідовність дій при обчисленні кратних інтегралів. Для реалізації цієї процедури перш за все потрібно мати m генераторів випадкових чисел, де m – дорівнює кратності інтегрованих.

Геометрично, обчислення m -кратного інтегралу

$$I = \int \int \dots \int_{(s)} f(x_1, x_2, \dots, x_m) dx_1 dx_2 \dots dx_m, \quad (6.20)$$

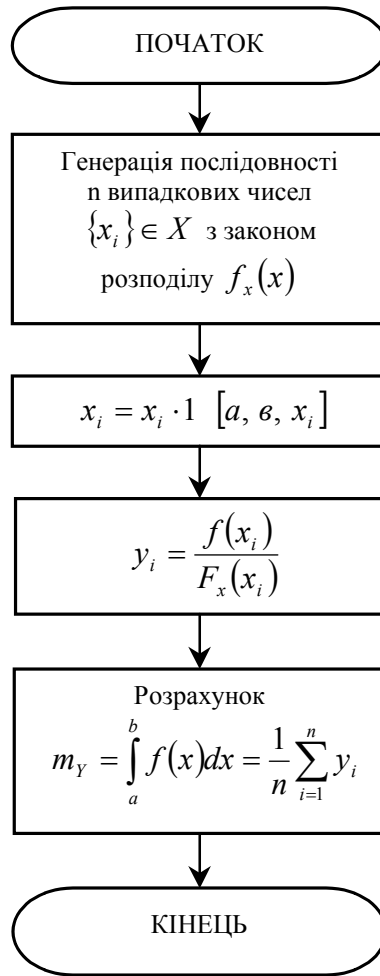


Рис. 6.5. Алгоритм методу Монте-Карло

де $y = f(x_1, x_2, \dots, x_m)$ – неперервна функція в обмеженій замкненій області S , зводиться до визначення $(m + 1)$ -вимірною обсягу прямого циліндра в просторі $0x_1, x_2, \dots, x_m, y$, що побудований на основі S й обмежений зверху поверхнею $y = f(x_1, x_2, \dots, x_m)$.

Для перетворення інтегралу (6.20) таким чином, щоб нова область інтегрування цілком знаходилась в середині одиничного m – вимірною куба σ , зробимо заміну змінних

$$x_i = a_i + (b_i - a_i)\xi_i,$$

де ξ_i – відповідні координати від 0 до 1; a_i, b_i – граничні значення координат, де розташована область інтегрування.

Тоді з (6.20) отримуємо

$$I = (a_1 - b_1)(a_2 - b_2) \dots (a_m - b_m) I_{\xi},$$

де

$$I_{\xi} = \int_{(\sigma)} \dots \int f(\xi_1, \xi_2, \dots, \xi_m) d\xi_1 d\xi_2 \dots d\xi_m.$$

Якщо застосувати m генераторів рівномірно розподілених випадкових чисел в діапазоні $(0,1)$, то обчислення середнього значення функції від їх комбінацій з застосуванням багатовимірного індикатора області інтегрування дасть шукану оцінку інтегралу

$$I_{\xi} = \frac{1}{n} \sum_{i=1}^n f(\xi_{1i}, \xi_{2i}, \dots, \xi_{mi}) 1[\xi_{1i}, \xi_{2i}, \dots, \xi_{mi}],$$

де $1[\xi_{1i}, \xi_{2i}, \dots, \xi_{mi}]$ дорівнює 1, якщо точка потрапляє в середину області інтегрування, і 0, якщо не потрапляє.

Похибка обчислення m -кратного інтегралу за методом Монте–Карло оцінюється аналогічно однократному за формулою (6.19).

6.5. Чисельне диференціювання

При розв'язанні багатьох практичних задач виникає необхідність визначення похідних від функцій, що задаються масивами даних, отриманих з експерименту. В цьому випадку безпосереднє диференціювання може дати безглузді результати, тому що не відокремлено шум, що дуже сильно впливає на похідну. Приклад зображено на рисунку 6.6, де $f(x)$ – корисний сигнал, $\tilde{f}(x)$ – сигнал, який виміряний (з шумом).

Зрозуміло, що похідна від $f(x)$ буде містити дуже сильну випадкову складову, що вносить велику похибку в визначення реальної похідної від X . Практичним шляхом розв'язання цієї проблеми є згладжування сигналу, що отриманий з експерименту, шляхом інтерполяції чи апроксимації, а далі диференціювання інтерполяційного полінома (чи апроксимувальної функції). При цьому для отримання похідних вищих порядків треба ставити умову їх збігання для сигналу та його інтерполяційного полінома. Похибка похідної інтерполяційної функції $P(x)$ дорівнює похідній від похибки цієї функції:

$$\Delta(x) = \tilde{f}(x) - P(x),$$

$$\Delta^*(x) = \tilde{f}'(x) - P'(x);$$

тобто внаслідок лінійності операцій диференціювання та віднімання маємо

$$\Delta^*(x) = \Delta'(x).$$

В довідниках існують спеціальні таблиці, що дозволяють знаходити з застосуванням різних різницевих інтерполяційних формул значення похідних. Взагалі такі формули нескладно отримати для будь-яких методів інтерполяції шляхом диференціювання інтерполяційних формул в загальному вигляді.

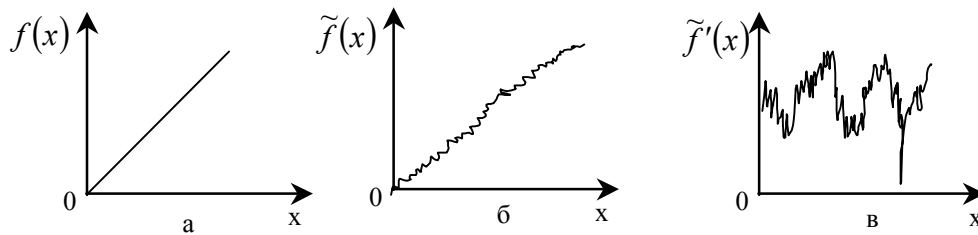


Рис. 6.6.

Контрольні запитання та завдання

1. Сформулюйте задачу інтерполяції.
2. В яких випадках інтерполяція неможлива?
3. З якою точністю можливо обчислити $\sqrt{2}$ за допомогою формули Лагранжа для функції \sqrt{x} , якщо вибрати вузли інтерполяції $x_0 = 81$, $x_1 = 1$, $x_2 = 4$?
4. Виведіть інтерполяційну формулу Ньютона для рівновіддалених вузлів.
5. Складіть алгоритм обчислення полінома Лагранжа.
6. Що таке сплайн – інтерполяція? Як визначаються коефіцієнти сплайнів?
7. Опишіть області застосування різницевих методів інтерполяції.
8. Що таке екстраполяція?
9. Якими шляхами розв'язується задача апроксимації?
10. Чим відрізняється апроксимація від інтерполяції?
11. Виведіть систему рівнянь для визначення коефіцієнтів апроксимувального полінома в методі найменших квадратів.
12. Які поліноми називаються ортогональними? Наведіть приклади.
13. Розробіть алгоритм та наведіть приклад програми для ЕОМ оцінки статистичних характеристик результатів вимірювання випадкової величини X .
14. Перелічіть методи чисельного інтегрування?
15. Чому необхідні методи і алгоритми чисельного інтегрування?

16. Яка послідовність застосування методів Ньютона-Котеса? Розробіть алгоритм. Чим відрізняються прості формули Ньютона-Котеса від складених?
17. Як отримують формули Ньютона-Котеса? Виведіть прості формули трапецій та Сімпсона для чисельного інтегрування та складені формули.
18. Як застосовуються методи Гаусса і Чебишева? Розробіть алгоритм.
19. Як визначаються коефіцієнти в формулі Гаусса для чисельного інтегрування? Чому її називають формулою найвищої алгебраїчної точності?
20. Як оцінюється похибка методів чисельного інтегрування?
21. Порівняйте методи Чебишева, Гаусса, Ньютона-Котеса?
22. Згадайте метод прямокутників. В чому його суть? Розробіть алгоритм.
23. Розробіть алгоритм та наведіть приклад програми чисельного інтегрування за методом Монте-Карло для двократного інтеграла

$$I = \iint_{(\sigma)} (x^2 + y^2) dx dy,$$

де область інтегрування σ визначається такими нерівностями

$$\begin{aligned} \frac{1}{2} &\leq x \leq 1, \\ 0 &\leq y \leq 2x - 1. \end{aligned}$$

Чи потраплять в область інтегрування точки з координатами (0.55; 0.75), (0.25; 0.75), (0.25; 0.25), (0.99; 0.70)?

24. Яким чином проводиться чисельне диференціювання даних, що отримані з експерименту та містять похибку?

Глава 7. Методи оптимізації і планування

Методи оптимізації широко застосовуються для розв'язання задач теорії оптимальних процесів, оптимального регулювання, вироблення керувальних збурень на об'єкти. Без розробки та застосування методів оптимізації неможливе керування ректифікаційними колонами в спиртовій промисловості, установками крекінгу нафти, конверторами при виробництві сталі та ін. До транспортних задач та задачі комівояжера зводяться багато задач економічної кібернетики (мережне планування, управління запасами, перевезеннями та ін.), керування організацією виробництва (розподіл завдань, обробка деталей, конвеєрне виробництво) та задачі оптимального програмування. Окрема група задач теорії оптимізації – це задачі оптимального проектування. Наприклад, задачі проектування радіоелектронних засобів з заданими обмеженнями на рівень шуму та смугу пропускання чи показниками надійності в умовах старіння.

7.1. Класична постановка задачі оптимізації

Звичайна постановка задачі оптимізації така: в деякому просторі S тим чи іншим засобом виділяється деяка непуста множина M точок цього простору, яку називають припустимою множиною. Далі фіксується деяка дійсна функція $f(x)$, що задана в усіх точках x допустимої множини. Вона називається цільовою функцією. Задача оптимізації полягає в тому, щоб знайти точку x_0 в множині M , для якої функція $f(x)$ приймає екстремальне (максимальне або мінімальне) значення. В першому випадку для всіх точок x множини M задовольняється нерівність $f(x_0) \geq f(x)$, в другому випадку – нерівність $f(x_0) \leq f(x)$.

В практичних задачах можливі дві основні постановки оптимізаційних задач. В першому випадку задача розглядається в звичайному (евклідовому) просторі кінцевої розмірності. Точками x допустимої множини будуть кортежі $x = (x_1, x_2, \dots, x_n)$ дійсних чисел, цільовою ж функцією $f(x) = F(x_1, \dots, x_n)$ буде звичайна дійсна функція від n дійсних аргументів (n – розмірність простору). Таку задачу ми будемо називати в подальшому задачею оптимізації функцій. В другому випадку постановки оптимізаційної задачі в якості припустимої множини виступає деяка множина M функцій дійсних змінних $y(x_1, \dots, x_m)$, а цільовою функцією є деякий функціонал F , який ставить у відповідність кожній функції $y(x_1, \dots, x_m)$ деяке дійсне число $F(y)$. Цю задачу ми будемо називати задачею оптимізації функціоналів або варіаційною задачею.

7.2. Класифікація задач оптимізації

Перш за все треба розділяти задачі параметричної та структурної оптимізації.

Параметрична оптимізація є предметом, що розглядається в цьому розділі, де наведені постановка такої задачі та методи її розв'язання. Структурна оптимізація – це задача синтезу оптимальної структури системи, причому зміна структур та перетворення однієї структури в іншу здійснюється за спеціальним алгоритмом синтезу. Параметрична оптимізація об'єднує багато різних задач, що мають свої власні особливості та методи розв'язання.

Класифікацію цих задач наведено на рисунку 7.1.

До цього треба додати деякий коментар:

1. Якщо існує декілька цільових функцій, то має місце задача векторної оптимізації.
2. Якщо кількість параметрів X , що управляються, більше ніж один, то розв'язується задача багатопараметричної оптимізації.
3. Якщо існують обмеження та умови, що зв'язують параметри X , то виникає задача оптимізації з умовами, яка в кібернетиці дістала назву математичного програмування.
4. Математичне програмування об'єднує задачі нелінійного програмування (цільова функція в загальному випадку нелінійна), стохастичного програмування (параметри X – випадкова величина, а цільова функція – випадкова функція), динамічного програмування (оптимізація багатокрокових процесів пошуку рішення).
5. Якщо параметри, що управляються, приймають тільки дискретні значення, то виникає задача дискретної оптимізації, а якщо X – цілі числа, то – задача цілочислового програмування.
6. У випадку, коли цільова функція опукла, та область, де задані X , теж опукла, то має місце задача опуклого програмування; якщо цільова функція та умови лінійні – лінійного (кусково-лінійного) програмування; цільова функція квадратична, а умови лінійні-квадратичного програмування; цільова функція та умови лінійні комбінації функцій однієї змінної – сепарабельного програмування; цільова функція та умови подані у вигляді поліномів – геометричного програмування.

7.3. Багатокритеріальна оптимізація

На практиці часто виникає випадок, коли замість однієї цільової функції $f(x)$ задано декілька цільових функцій $f_1(x), \dots, f_R(x)$. Така задача багатокритеріальної оптимізації має декілька постановок. В одній з них



Рис. 7.1. Класифікація задач оптимізації

потрібно оптимізувати один з критеріїв, припустимо, $f_1(x)$, причому решту критеріїв утримують в заданих межах: $a_i \leq f_i(x) \leq b_i$ ($i = 2, 3, \dots, k$). В цьому разі фактично йдеться про звичайну багатокритеріальну оптимізацію. Що ж до нерівностей, які обмежують інші критерії, то їх

можна розглядати як додаткові обмеження на припустиму область M . В другому випадку постановка полягає в упорядкуванні заданої множини критеріїв та послідовній оптимізації за кожним з них. Інакше, якщо проводять оптимізацію за першим критерієм $f_1(x)$, то одержують деяку множину $M_1 \subset M$, на якій функція $f_1(x)$ приймає оптимальне (екстремальне) значення. Приймавши його за нову допустиму множину, проводять оптимізацію за другим критерієм та одержують в результаті нову допустиму множину $M_2 \subset M_1$. Якщо продовжити цей процес, то можна одержати після оптимізації за останнім критерієм $f_R(x)$ множину M_R , яка і буде кінцевим результатом багатокритеріальної оптимізації. Звідси, якщо на деякому кроці i ($i < k$) множина M_1 зведеться до однієї точки, процес оптимізації можна буде закінчити, оскільки $M_i = M_{i+1} = \dots = M_k$. Зрозуміло, що як і в випадку звичайної однокритеріальної оптимізації, задача може взагалі не мати розв'язку.

Третя постановка застосовує процес зведення багатьох критеріїв до одного за рахунок введення апріорних вагових коефіцієнтів λ_i для кожного з критеріїв $f_i(x)$. В якості таких коефіцієнтів можуть бути вибрані будь-які дійсні числа. Їх значення вибирають, виходячи з інтуїтивного подання ступеня важливості різних критеріїв: більш важливі критерії одержують ваги з більшими абсолютними значеннями. Після встановлення ваг λ_i багатокритеріальна задача зводиться до однокритеріальної з цільовою функцією $f(x) = \lambda_1 f_1(x) + \dots + \lambda_R f_R(x)$.

Замість простої лінійної комбінації вхідних критеріїв можуть використовуватися і більш складні засоби формування з них нового критерію.

7.4. Гладка оптимізація

У випадку, коли функція цілі $f(x)$ і функції $p_i(x)$, які задають обмеження, є диференціальними (гладкими), для розв'язання задач оптимізації використовується поняття градієнта. Для будь-якої функції $g(x)$, що диференціюється, її градієнтом $\nabla g(x)$ в точці x називається вектор

$$\nabla g(x) = \begin{pmatrix} \frac{\partial g(x)}{\partial x_1} \\ \dots \\ \frac{\partial g(x)}{\partial x_n} \end{pmatrix}.$$

Вектор градієнта $\nabla g(x)$ задає (в даній точці x) напрям найшвидшого росту функції $g(x)$, а зворотний йому напрям $-\nabla g(x)$, що називається антиградієнтом, – напрям найшвидшого спадання цієї функції. Точки, в яких градієнт функції перетворюється в нуль, називаються її стаціонарними точками. Якщо екстремум функції $f(x)$ досягається всередині припустимої області (не на її межі), то в точці оптимуму $x = a$ її градієнт перетворюється в нуль, тобто має місце система рівнянь

$$\left. \frac{\partial f(x)}{\partial x_1} \right|_{x=a} = 0, \dots, \left. \frac{\partial f(x)}{\partial x_n} \right|_{x=a} = 0.$$

Ці рівняння (умови стаціонарності функцій) мають місце не тільки для абсолютних, але й для локальних екстремумів. Разом з тим умови стаціонарності не є достатніми.

7.4.1. Умови Куна-Таккера

У випадку, коли екстремальна точка a лежить на межі припустимої множини, вона не обов'язково є стаціонарною. Але за деякої додаткової умови може, якщо замінити цільову функцію $f(x)$ так званою функцією

Лагранжа $L(x) = f(x) + \sum_{i=1}^m \lambda_i p_i(x)$, (де $p_i(x)$ – ліві частини всіх граничних умов), дістати того, щоб граничні екстремальні точки функції $f(x)$ були стаціонарними точками функції Лагранжа $L(x)$ при відповідних значеннях параметрів λ_i .

Умова, про яку йдеться, – так звана умова регулярності – виконується на практиці. Вона полягає в лінійній незалежності в даній точці $x = a$ градієнтів всіх мережних функцій $p_i(x)$, для яких $p_i(a) = 0$.

При виконанні цієї умови для будь-якої точки максимуму x^* в задачі “знайти $\max f(x)$ при $p_i(x) = 0, (i = 1, \dots, l), p_i(x) \geq 0, (i = l + 1, \dots, m)$ ” і для будь-якої точки мінімуму x^* в задачі “знайти $\min f(x)$ при $p_i(x) = 0, (i = 1, \dots, l), p_i(x) \leq 0, (i = l + 1, \dots, m)$ ” повинні виконуватися такі три умови, які називаються умовами Куна-Таккера:

- 1) x^* лежить в припустимій множині;
- 2) $\lambda_i p_i(x^*) = 0$ при $i = 1, \dots, m$;
- 3) $\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla p_i(x^*) = 0$.

Тут λ_i – деякі дійсні числа (множники Лагранжа), довільні для всіх умов типу рівності ($i = 1, \dots, l$) і невід'ємні для всіх умов типу нерівності ($i = l + 1, \dots, m$).

7.4.2. Чисельні методи гладкої оптимізації

Метод оптимізації, який використовує умови Куна-Таккера, на практиці застосовується відносно рідко через велику складність аналізу системи співвідношень, що виникає. Найчастіше застосовуються чисельні методи оптимізації, для яких достатньо вміти знаходити чисельні значення градієнта в будь-якій заданій точці.

Загальна ідея методу градієнтного спуску (підйому)

Нехай неперервно диференційовна цільова функція $f(x)$ задана в усіх точках простору X . Вирушуючи від точки x на дуже малий крок ε в будь-якому напрямку d , де d – одиничний вектор, в силу умови диференційовності функції $f(x)$, одержимо нерівності:

$$f(x + \varepsilon d) \geq f(x) \geq f(x - \varepsilon d).$$

Це означає, що рух (на дуже малий крок) в напрямку градієнта функції забезпечує найбільший підйом, а в напрямку антиградієнта – найбільше спадання цієї функції. Ці напрямки називають, відповідно, напрямком найшвидшого підйому і найшвидшого спуску функції $f(x)$ в даній точці x .

Виходячи з заданої точки x^0 , будемо послідовність точок (x^0, x^1, x^2, \dots) так, що переміщення від кожної точки x^{i-1} до наступної точки x^i проводиться в напрямку найшвидшого підйому (при пошуку максимуму) або найшвидшого спуску (при пошуку мінімуму).

Важкість практичного застосування такої процедури полягає перш за все у виборі величини кроку ε_i при переході від точки x^{i-1} до точки x^i . Справа в тім, що нерівності мають місце лише в малому околі точки x , тобто для малої величини кроку ε_i . При великому значенні кроку можна, прямуючи в напрямку найшвидшого підйому, одержати не збільшення, а зменшення значення цільової функції $f(x)$. Те ж саме має місце і для процедури спуску.

Отже, величину кроку ε_i в процедурах найшвидшого підйому і спуску необхідно вибирати достатньо малою. Проте при прямованні малими кроками для наближення до екстремальної точки може знадобитись дуже велика кількість кроків. Крім того, від процедур підйому

і спуску звичайно потребується не просто наближення до екстремальної точки $x = a$, а збіжність до неї. Це означає, що зі збільшенням кількості кроків i відстань $|x^i - a|$ від точки x^i до точки екстремуму $x = a$ повинна наближатися до нуля. Це можливо лише в тому випадку, якщо з наближенням до точки екстремуму величина кроку ε_i буде наближатись до нуля. На рисунку 7.2 наведений алгоритм пошуку екстремуму функції за методом градієнтного спуску.

Пропорційно-градієнтний метод

Найбільш простим способом забезпечення необхідного зменшення кроку при наближенні до точки екстремуму для диференціальної функції $f(x)$ є вибір довжини кроку ε_i пропорційним довжині вектора градієнта в точці x^{i-1} :

$$\varepsilon_i = \alpha |\nabla f(x^{i-1})|, \quad \alpha > 0, \quad \alpha = const.$$

Такий вибір кроку означає, що перехід від точки x^{i-1} до точки x^i буде виконуватись за формулою $x^i = x^{i-1} + \alpha \nabla f(x^{i-1})$ для процедури підйому (знаходження максимуму функції $f(x)$) і за формулою $x^i = x^{i-1} - \alpha \nabla f(x^{i-1})$ для процедури спуску (знаходження мінімуму функції $f(x)$).

Повнокроковий градієнтний метод

В цьому методі кожний крок градієнтного підйому або спуску виконується максимально можливою довжиною, яка забезпечує необхідний напрямок зміни значення функції (тобто її збільшення або зменшення). Інакше кажучи, на напівпрямій, що виходить з чергової точки x^{i-1} в напрямку градієнта (при підйомі) або антиградієнта (при спусканні) відшукується точка абсолютного максимуму або мінімуму, яка і вибирається як наступна точка x^i .

В випадку довільної диференціальної функції $f(x)$ розв'язання подібної задачі, що її називають одновимірною оптимізаційною задачею, спрощується при накладанні на функцію $f(x)$ деяких додаткових вимог, наприклад, опуклості.

Метод спряжених градієнтів

Для випадку, коли функція $f(x)$ – квадратний поліном, розроблена точна процедура, при повних кроках якої точки екстремуму досягаються з будь-якої початкової точки x^0 за n кроків (де n – розмірність простору).

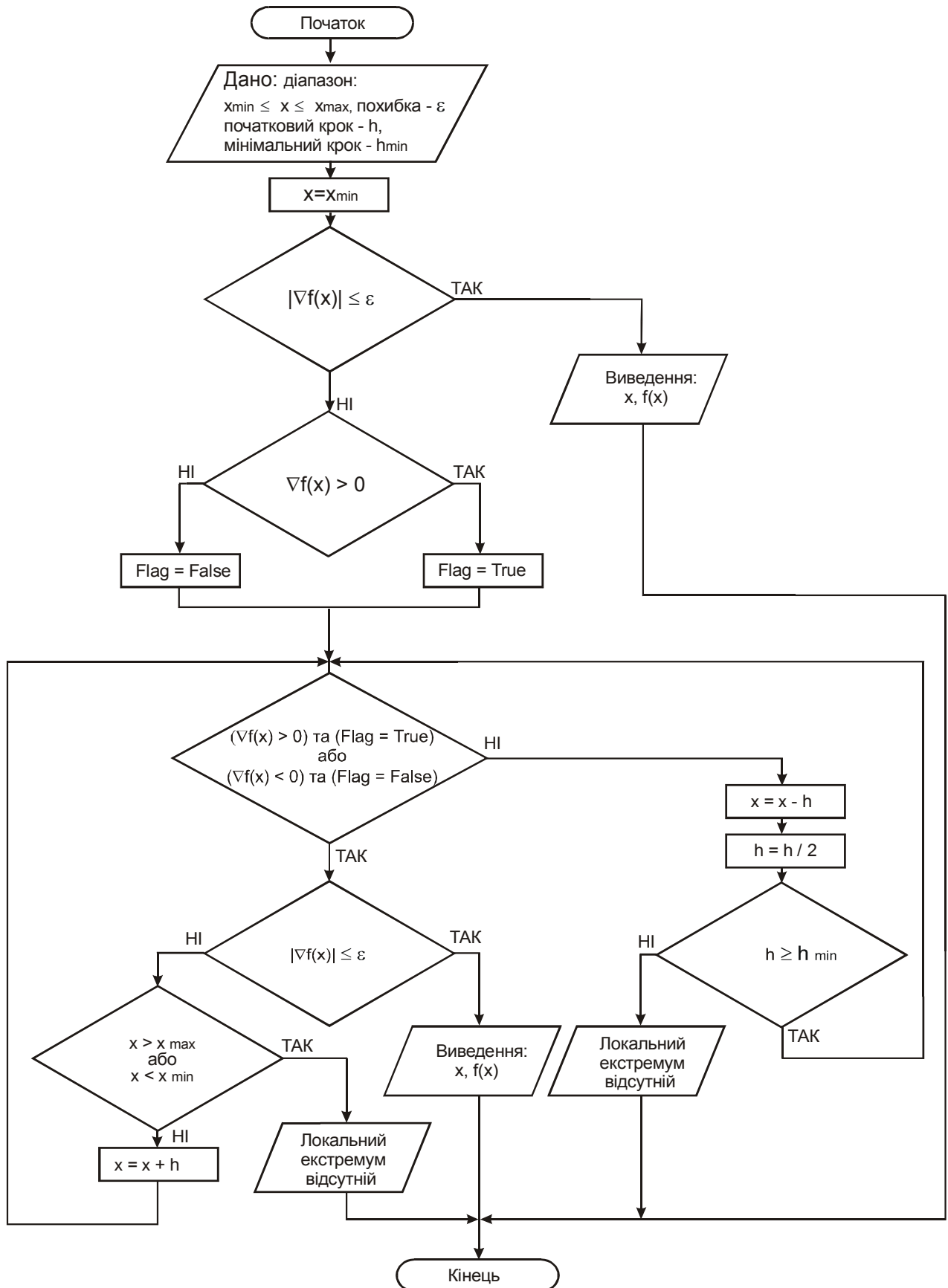


Рис. 7.2. Алгоритм методу градієнтного спуску (підйому)

Напрямок α^{i+1} кроку від точки x^i до точки x^{i+1} задається в цьому методі такою рекурентною формулою:

$$a^{i+1} = \pm \nabla f(x^i) + \frac{|\nabla f(x^i)|^2}{|\nabla f(x^{i-1})|^2} a^i, \quad i = 1, 2, \dots,$$

де $|\nabla f(x)|$ означає довжину вектора $\nabla f(x)$.

Знак плюс вибирається в тому випадку, коли розглядається задача максимізації, а знак мінус – в випадку розв’язання задачі мінімізації. Такий же вибір знака здійснюється і для напрямку α^1 першого кроку ($i = 0$). Він збігається з напрямком $\pm \nabla f(x^0)$.

7.4.3. Методи зведення загальної задачі оптимізації до задачі без обмежень

Чисельні методи гладкої оптимізації, які описані вище, в чистому вигляді застосовуються звичайно для задач без обмежень. Задачі загального вигляду (тобто задачі з обмеженнями) зводять до задач без обмежень за рахунок зміни цільової функції. Нехай, наприклад, потрібно знайти максимум функції $f(x)$ при обмеженнях $p_i(x) \geq 0$ ($i = 1, \dots, m$). В так званому методі штрафних функцій будується функція $P(x)$, яку називають штрафною функцією, і яка всередині припустимої області M приймає нульове значення, а поза нею – від’ємне. Найчастіше в якості такої функції вибирають функцію вигляду

$$P(x) = -\sum_{i=1}^m |\min[p_i(x), 0]|^l, \quad l \geq 1,$$

після чого будують послідовність додатних чисел $r_1 > r_2 > \dots > r_k > \dots$, яка збігається до нуля.

При заміні цільової функції $f(x)$ функцією

$$F_k(x) = f(x) + \frac{1}{r_k} P(x), \quad k = 1, 2, \dots,$$

розв’язують задачу без обмежень для цієї функції, починаючи з $k = 1$. Якщо розв’язок x^k належить припустимій області, тобто якщо всі $g_i(x^k) \geq 0$, то x^k є розв’язком початкової задачі з обмеженнями. В

протилежному випадку замінюють k на $k+1$ і розв'язують задачу для нової цільової функції $F_k(x)$.

Алгоритм розв'язання такої задачі наведений на рисунку 7.3.

В методі бар'єрів при розв'язанні задачі $\max f(x)$ при $p_i(x) \geq 0$ ($i=1, \dots, m$) цільову функцію $f(x)$ замінюють функцією $F_k(x) = f(x) + r_k B(x)$, $r_k > 0$, де бар'єрна функція $B(x)$ характеризується властивістю прямувати до $-\infty$ при наближенні до границі припустимої області M зсередини. Таку функцію можна задати, наприклад, формулою:

$$B(x) = -\sum_{i=1}^m \frac{1}{p_i(x)},$$

якщо область M задана нерівностями $p_i(x) \geq 0$ ($i=1, \dots, m$).

Таким же чином, як і в методі штрафних функцій, вибирають послідовність $\{r_k\}$ додатних чисел $r_1 > r_2 > \dots > r_k > \dots$, яка збігається до нуля. Розв'язуються задачі без обмежень для цільових функцій $F_k(x) = f(x) + r_k B(x)$ при $k=1, 2, \dots$

Якщо при цьому використовується той або інший чисельний метод, в якому початкова точка вибирається всередині M , то наявність бар'єра забезпечить належність екстремальної точки x^k (при її існуванні) області M . Наявність екстремуму забезпечується умовами обмеженості в області M і неперервності функцій $f(x)$ і $B(x)$.

7.5. Опукла оптимізація

Задача оптимізації може бути значно спрощена при накладанні на цільову функцію $f(x)$ і припустиму область M деяких обмежень. Одним з таких обмежень є так звана умова опуклості.

Множина M евклідового простору називається опуклою, якщо для будь-яких двох точок x^1 і x^2 цієї множини всі точки відрізка $[x^1, x^2]$, що їх з'єднує, також належать множині M .

Функція $f(x)$ опукла, або, точніше, опукла донизу, якщо вона визначена на опуклій множині M і для будь-яких двох точок x^1 і x^2 цієї множини значення $f(x)$ функції в будь-якій точці x відрізка $[x^1, x^2]$ не перевищують значення в тій же точці, що визначена на даному відрізку лінійної функції із значеннями $f(x_1)$ і $f(x_2)$ на його кінцевих точках. Це, в загальному випадку, виражається нерівністю

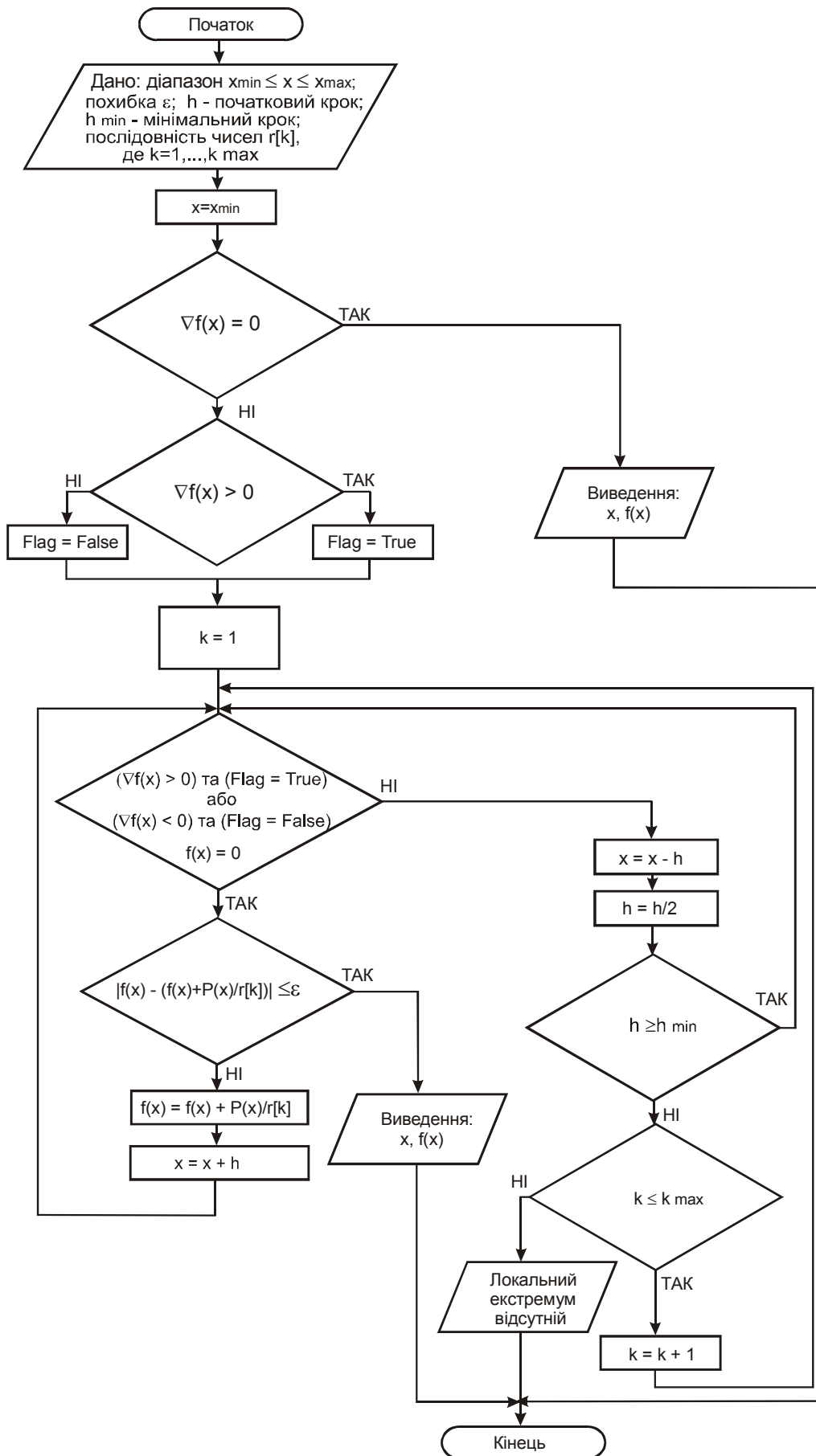


Рис. 7.3. Алгоритм методу штрафних функцій

$$\alpha f(x^1) + (1 - \alpha)f(x^2) \geq f(\alpha x^1 + (1 - \alpha)x^2) \text{ при } 0 \leq \alpha \leq 1.$$

Очевидно, що при $0 \leq \alpha \leq 1$ вираз $\alpha x^1 + (1 - \alpha)x^2$ задає різні точки відрізка $[x^1, x^2]$.

Для опуклої функції $f(x)$ в будь-якій точці x^0 , що лежить в області визначення M функції, існує вектор $r(x^0)$, для якого при будь-якому x і M виконується нерівність $f(x) - f(x^0) \geq r(x^0);(x - x^0)$.

В правій частині цієї нерівності стоїть скалярний добуток векторів $r(x^0)$ і $x - x^0$, тобто добуток довжин цих векторів на косинус кута між ними. Вектор $r(x^0)$, що задовольняє цю властивість, називається узагальненим градієнтом (субградієнтом) функції $f(x)$ в точці x^0 . Звичайний градієнт (у випадку його існування і відмінності від нуля) є також узагальненим градієнтом, причому єдиним, в розглядуваній точці. Якщо функція $f(x)$ не диференційовна в деякій точці x^0 , в ній існує деяка множина узагальнених градієнтів, яку називають субградієнтною множиною функції $f(x)$ в точці x^0 .

7.5.1. Простий субградієнтний метод опуклої оптимізації

Для реалізації субградієнтної оптимізації в випадку задачі без обмежень процес може починатися з будь-якої точки x^0 . Зсув чергової точки x^i до наступної точки x^{i+1} здійснюється на відстань l_i в напрямку будь-якого узагальненого градієнта функції $f(x)$ в точці x^i в випадку задачі максимізації вгнутої функції і в зворотному напрямку – в випадку задачі мінімізації опуклої функції. Якщо кроки l_i зсувів обираються таким чином, що $l \rightarrow 0$ при $i \rightarrow \infty$, а ряд $l_0 + l_1 + l_2 + \dots$ розбігається, то послідовність $\{x^0, x^1, \dots, x^m, \dots\}$ збігається до множини екстремуму заданої функції $f(x)$ за умови, що множина обмежена.

Для обчислення субградієнтів застосовуються різні способи. Одним з них є зведення задачі обчислення субградієнта функції, що задана складним виразом, до обчислення субградієнтів окремих компонентів цього виразу. Для подібного зведення особливо часто застосовуються такі властивості опуклих функцій.

Властивість 1. Якщо функції $f_1(x), \dots, f_m(x)$ опуклі, то опуклою буде і будь-яка їх лінійна комбінація $f(x) = \sum_{i=1}^m a_i f_i(x)$ з невід'ємними коефіцієнтами a_i ($a_i \geq 0, i = 1, \dots, m$), а субградієнт $S(x)$ функції $f(x)$ дорівнює лінійній комбінації $\sum_{i=1}^m a_i S_i(x)$ субградієнтів $S_i(x)$ функцій $f_i(x)$.

Властивість 2. Якщо функції $f_1(x), \dots, f_m(x)$ опуклі, то опуклою буде і функція $f(x) = \max f_i(x)$, і всі субградієнтні множини $S_i(x^0)$ функцій $f_i(x)$, для яких $f(x^0) = f(x)$, входять в субградієнтну множину $S(x^0)$ функції $f(x)$ в будь-якій заданій точці $x = x^0$.

7.5.2. Методи розтягу простору

Простий субградієнтний метод, що розглянутий в попередньому розділі, може виявитися повільно збіжним, якщо області рівня цільової функції сильно витягнуті в одному напрямку, або, як часто при цьому говориться, мають вигляд вузьких і довгих ярів. Ця ситуація характерна для багатьох задач негладкої оптимізації. Для поліпшення збіжності методу в подібних випадках використовують методи розтягу простору.

Розтяг простору зводиться до заміни змінних x_1, \dots, x_k в цільовій функції $f(x) = f(x_1, \dots, x_n)$ деякими їх лінійними комбінаціями. Особливо просто виконується розтяг в напрямку однієї з координатних осей. Змінна x_i , що відповідає цій осі, замінюється при цьому на $k^{-1}x_i$, де k – коефіцієнт розтягу, а решта залишається без змінювання.

7.6. Негладка оптимізація за методом координатного спуску (підйому)

При оптимізації недиференційовних функцій можна відмовитись від будь-яких узагальнень понять градієнта та використовувати лише кроки вздовж осей координат. З цією метою досліджують значення цільової функції $f(x_1, \dots, x_n)$ на черговому кроці оптимізації з приростом $\pm \delta_i$ однієї з координат і переходять від точки (x_1, \dots, x_n) до точки $(x_1, \dots, x_{i-1}, x_i \pm \delta_i, x_{i+1}, \dots, x_n)$. Знак приросту вибирають таким чином, щоб значення функції $f(x_1, \dots, x_n)$ змінювалось в потрібному напрямку (збільшувалось при максимізації і зменшувалось при мінімізації).

Величини δ_i вибирають звичайно як в покрокових методах та забезпечують максимально можлива зміна функції в обраному напрямку.

Зсуви здійснюються по черзі по всіх осях координат, причому після зсуву по останній осі x_n знову повертаються до першої осі x_1 . Якщо повторити цей процес достатньо багато разів, як правило, можна знайти як завгодно точне наближення деякого екстремуму заданої негладкої функції в тому разі, коли виконується припущення, що такий екстремум існує.

7.7. Стохастична оптимізація

Іноді корисно замінити зсуви вздовж координатних осей зсувами (з відповідним знаком) вздовж випадково вибраного на кожному кроці напрямку. При достатньо загальних припущеннях внаслідок таких зсувів з імовірністю одиниця за достатньо велике число кроків екстремальна точка може бути апроксимована з будь-яким необхідним ступенем точності. Перевагою методу є простота кожного окремого кроку оптимізації. Проте кількість кроків порівняно з методами, які забезпечують пошук задовільних напрямків руху, у подібних простих методів, як правило, значно більша. При цьому на практиці вони можуть виявитися менш ефективними, ніж більш складні методи, що використовують поняття субградієнта і різні додаткові способи прискорення збіжності.

7.8. Лінійне програмування

Однією з задач опуклої оптимізації, що найчастіше зустрічаються, є так звана задача лінійного програмування, в якій як цільова функція, так і всі обмеження лінійні. Для розв'язання цієї задачі можуть бути застосовані загальні способи опуклої оптимізації, які описані в п'ятому розділі цієї глави. Але на практиці для цього випадку застосовують інші, більш прості способи, з одним з яких (так званий симплекс-метод) ми зараз ознайомимося.

Постановки задач лінійного програмування, які зустрічаються на практиці, передбачають ще одне додаткове обмеження. Це умова невід'ємності значень всіх змінних. Далі, обмеження типу нерівностей $p_i(x) \geq 0$ або $p_i(x) \leq 0$ введенням додаткових змінних з невід'ємними значеннями перетворюються в рівності $p_i(x) - z_i = 0$ або $p_i(x) + z_i = 0$. Крім того, простою зміною знака цільової функції задача знаходження максимуму зводиться до задачі знаходження мінімуму. Таким чином, в лінійному програмуванні можна обмежитись лише розв'язанням задачі мінімізації.

Стандартна постановка задачі лінійного програмування полягає в тому, щоб: знайти мінімум лінійної функції $d + c_1x_1 + c_2x_2 + \dots + c_nx_n$ при обмеженнях $b_i + a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = 0$ ($i = 1, \dots, m$), $x_i \geq 0$ ($i = 1, \dots, m$).

Інакше, йдеться про знаходження невід'ємного розв'язку $x = (x_1, \dots, x_n)$ даної системи лінійних алгебраїчних рівнянь

$b_i + \sum_{j=1}^n a_{ij}x_j = 0$ ($i = 1, \dots, m$), яке перетворює в мінімум значення даної

лінійної функції $d + \sum_{j=1}^n c_jx_j$.

Задача лінійного програмування може не мати розв'язку в таких випадках:

1) система S рівнянь $b_i + \sum_{j=1}^n a_{ij}x_j = 0$ ($i = 1, \dots, m$) несумісна, тобто взагалі

не має розв'язків;

2) система S не має жодного від'ємного розв'язку;

3) на множині M невід'ємних розв'язків системи S цільова функція

$f = d + \sum_{j=1}^n c_jx_j$ може приймати як завгодно великі за абсолютною

величиною від'ємні значення, тобто $\min_M f = -\infty$.

7.8.1. Симплекс-метод

Один з найбільш розповсюджених методів розв'язування задачі лінійного програмування в стандартній постановці полягає в послідовному застосуванні до лінійних функцій так званих симплексних перетворень, що

являють собою розв'язки одного з обмежувальних рівнянь $b_i + \sum_{j=1}^n a_{ij}x_j = 0$

відносно якого-небудь невідомого x_j , і підстановку знайденого таким чином значення x_j до всіх обмежувальних рівнянь і в цільову функцію

$f = d + \sum_{j=1}^n c_jx_j$.

7.8.2. Транспортна задача

В деяких окремих випадках розв'язання задачі лінійного програмування може бути значно спрощено. Найважливішим з таких

випадків є так звана транспортна задача, яку називають іноді задачею про призначення. Суть її полягає в мінімізації цільової функції

$$f = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

при $m + n$ обмеженнях типу рівнянь

$$\sum_{j=1}^n x_{ij} = a_i, \quad \sum_{i=1}^m x_{ij} = b_j.$$

Звичайно додатково припускається зберігання рівняння

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j,$$

інакше задача не буде мати розв'язків.

Назву транспортної задача, що розглядається, одержала тому, що до неї зводиться оптимізація плану перевезень вантажів з m пунктів відправлення з запасами a_1, \dots, a_m до n пунктів призначення з об'ємами споживання b_1, \dots, b_n . Роль коефіцієнтів c_{ij} в цільовій функції виконують питомі вартості, тобто вартості перевезення однієї одиниці вантажу з пункту i в пункт j . Задача полягає в мінімізації загальної вартості перевезення вантажів за умови, що вантажі повністю вивезені з усіх пунктів відправлення і потреби всіх пунктів призначення виявились повністю задоволеними.

7.8.3. Цілочислове лінійне програмування

Іноді на практиці доводиться зустрічатися з такими задачами лінійного програмування, в яких припустимі лише цілочислові розв'язки.

В загальному випадку одержання цілочислового розв'язку потребує застосування спеціальних методів, серед яких найчастіше вживають методи відтинань та розгалужень і границь.

Методи відтинань базуються на таких ідеях. Спочатку розв'язується звичайна задача лінійного програмування A_0 (з тимчасово відкинутою вимогою цілочислового розв'язку). Якщо координати точки $a^0 = (a_1, \dots, a_n)$, яка одержана в результаті розв'язання задачі A_0 , цілі числа, то ця точка дає розв'язок не тільки задачі A_0 , але й початкової

задачі цілочислового лінійного програмування, яку ми домовимося позначати через A .

Якщо хоча б одна з координат точки a^0 , наприклад, a_j , не є цілим числом, то до початкової задачі цілочислового лінійного програмування додається нове обмеження, що побудоване з використанням інформації, яка є в таблиці симплекс-методу і відповідає кроку, на якому побудований розв'язок a^0 .

В результаті виходить нова задача A_1 , до якої знов застосовується та ж сама процедура. Таким чином, методи відтинань будують і розв'язують послідовність лінійних задач A_0, A_1, \dots , кожна з яких відрізняється від попередньої лише одним новим обмеженням.

7.8.4. Загальна задача лінійної оптимізації

В загальному випадку задача дискретної оптимізації не припускає лінійності як цільової функції $f(x)$, так і обмежень $p_i(x) \geq 0$ ($i = 1, \dots, m$). Не вимагається також неодмінно цілочисловість координат точок, серед яких знаходиться розв'язок. Важливо лише, щоб вони утворювали дискретну множину M . Звичайно на практиці загальне число N точок виявляється настільки великим, що простий перебір фізично неможливий навіть при використанні найпотужніших ЕОМ. З цієї причини основна задача в дискретній оптимізації зводиться до розробки методів, які направлені на максимально можливе звуження перебору.

В так званих прямих методах дискретної оптимізації звичайно використовуються ті чи інші аналоги (в неперервному випадку) градієнтних методів, що розглядалися вище. До них відносять методи локальної оптимізації, наприклад, метод вектора спаду. Обчислювальну схему цього методу для розв'язання задачі мінімізації дійсної функції $f(x)$, що визначена на деякому просторі M дискретного метричного простору D , можна описати таким чином.

Спочатку вибираємо деяке початкове наближення $x^0 \in M$ і радіус r . В просторі D розглядаємо окіл $O(x^0, r)$ радіуса r з центром в точці x^0 . Досліджуючи координати вектора спаду, який характеризує зміни значень цільової функції в точках множини $O(x^0, r) \cap M$ порівняно з $f(x^0)$, визначаємо, чи є x^0 точкою локального мінімуму функції f . Якщо ні, то за допомогою вектора спаду в множині $O(x^0, r) \cap M$ вибираємо точку x^1 , для якої значення цільової функції менше, ніж $f(x^0)$. На наступній ітерації знову повторюємо цю ж процедуру, виходячи вже з x^1 як з центра нового околу, і так далі. Процес обчислень вважається закінченим, якщо

одержаний деякий локально оптимальний розв'язок задачі. Алгоритм методу припускає переривання обчислень на будь-якій ітерації з видачею в якості результату останнього одержаного припустимого розв'язку задачі.

В релаксаційних методах звертаються до способу ослаблення (релаксації) обмежень і заміни цільової функції $f(x)$ її мінорантою $f'(x)$ (в задачі мінімізації), тобто функцією, яка задовольняє умову $f'(x) \leq f(x)$ для всіх x .

З припустимої області M' , що здобута розширенням вихідної області $M (M' > M)$ за рахунок ослаблення обмежень, M' і $f'(x)$ вибираються так, щоб, по-перше, релаксована задача допускала порівняно прості способи розв'язувань, і, по-друге, щоб вона найменше відрізнялась від вихідної за цільовою функцією і множиною припустимих розв'язків:

$$\min_{x \in M'} f'(x) \leq \min_{x \in M'} f(x) \leq \min_{x \in M} f(x)$$

Метод розгалужень і границь (МРГ)

Цей метод дозволяє отримати точний або наближений розв'язок з заданою відносною похибкою за цільовою функцією. Суть МРГ полягає в тому, що замість вихідної задачі оптимізації $A_0 = \{\min f_0(x), x \in M_0\}$ будується і розв'язується послідовність релаксованих задач $\bar{A}_i = \{\min \bar{f}_i(x), x \in \bar{M}_i\}$, $i = 0, 1, 2, \dots$. Спочатку розв'язується задача для M_0 і $\bar{f}_0(x_0) = f_0(x_0)$, то x_0 – оптимальний розв'язок A_0 , і процес закінчується. Інакше $\bar{f}_0(x_0)$ – оцінка знизу для цільової функції на оптимальному розв'язку задачі A_0 , і алгоритм починає процес розгалуження вихідної задачі A_0 на декілька задач (безпосередніх нащадків).

7.9. Теорія ігор

До методів дискретної оптимізації і лінійного програмування часто доводиться звертатись при аналізі і виборі рішень в конфліктних ситуаціях, тобто при наявності сторін, що мають на меті відмінні (найчастіше – протилежні) цілі.

Стратегією гравця називають сукупність правил, що визначають поведінку гравця від початку гри до її завершення.

Задання стратегій (A, B) двох гравців в парній грі повністю визначає її наслідок, тобто виграш одного і програш другого. Гра називається скінченною, якщо в кожного гравця є лише скінченна кількість стратегій. Результати скінченної парної гри з нульовою сумою можна задавати матрицею, рядки і стовпці якої відповідають відмінним стратегіям, а її

елементи – відповідні виграші однієї сторони (які дорівнюють програшам другої).

Розглянемо гру $m \times n$ з матрицею

	B_1	B_2	...	B_n
A_1	a_{11}	a_{12}	...	a_{1n}
A_2	a_{21}	a_{22}	...	a_{2n}
...
A_m	a_{m1}	a_{m2}	...	a_{mn}

Якщо перший гравець застосовує стратегію A_i , то другий буде прагнути до того, щоб вибором відповідної стратегії B_j звести виграш першого гравця до мінімуму. Величина цього мінімуму, яку ми позначимо α_i , дорівнює, очевидно, $\min_j a_{ij}$.

З точки зору першого гравця (при будь-яких відповідях супротивника) доцільно прагнути знайти таку стратегію, при якій α_i перетворюється в максимум. Цей максимум, який ми позначимо α , називається нижньою ціною гри. Оскільки значення α обчислюється за формулою $\alpha = \max_i \min_j a_{ij}$, то його називають також максиміном. Йому відповідає максимінна стратегія, дотримуючись якої, перший гравець при будь-яких стратегіях у противника забезпечить собі виграш, який не менше α (в залежності від знака α це може бути і програш, який в цьому разі виявиться мінімальним).

Аналогічно визначається мінімальний програш (який може бути в дійсності і виграшем) для другого гравця: $\beta = \min_j \max_i a_{ij}$.

Величина β називається верхньою ціною гри або мінімаксом. Їй відповідають мінімаксні стратегії другого гравця.

Має місце нерівність $\alpha \leq \beta$. При $\alpha = \beta$ розв'язок одержується в чистих стратегіях. Для знаходження їх достатньо виділити в платіжній матриці максимінні (тобто рівні α) елементи. Будь-яка пара рядків і стовпців, на перетині яких знаходиться такий елемент, відповідає парі чистих оптимальних стратегій. При $\alpha < \beta$ перший гравець може істотно збільшити свій середній виграш порівняно з α , якщо він буде користуватись не чистою (однією єдиною) стратегією, а так званою змішаною стратегією. Змішана стратегія C полягає в тому, що при повторі гри здійснюється випадковий вибір стратегії з деякої множини змішуваних стратегій, і для кожної змішуваної стратегії вказується імовірність її вибору.

7.10. Динамічне програмування

На практиці часто доводиться зустрічатись з випадками, коли метою (ціллю) оптимізації є встановлення найкращої послідовності тих чи інших робіт (виробничих операцій, етапів будівництва різних споруд тощо). З подібною метою зустрічаються при розв'язанні задач так званого динамічного програмування.

Однією з перших задач такого роду, що привернули увагу математиків, була так звана задача про комівояжера (мандрівного торговця). Суть її така: є $n + 1$ міст A_0, A_1, \dots, A_n ($n \geq 1$) з заданими між ними відстанями d_{ij} ($i, j = 0, 1, \dots, n$). Потрібно, відправившись з A_0 , вибрати такий маршрут пересування $A_0, A_{i_1}, A_{i_2}, \dots, A_{i_n}, A_0$, при якому комівояжер, побувавши в кожному місті по одному разу, повернувся б до вихідного пункту A_0 , пройшовши при цьому мінімально можливий сумарний шлях.

Основний спосіб динамічного програмування полягає в знаходженні правил домінування, які дозволяють робити порівняння варіантів розвитку послідовностей і завчасне відсіювання безперспективних варіантів. У ряді випадків в задачах динамічного програмування вдається одержати такі сильні правила домінування, що вони визначають елементи оптимальної послідовності однозначно один за одним. В такому випадку правила домінування називають розв'язувальними правилами.

Розв'язувальні правила звичайно виводяться за допомогою принципу оптимальності Беллмана. Суть принципу оптимальності така. Нехай критерій F (задається формулою або алгоритмом), який дає числову оцінку якості варіанта (послідовності) $A_n = A_{i_1}A_{i_2}\dots A_{i_n}$, можна застосовувати не тільки до всієї послідовності, але і до будь-якого її початкового відрізка $A_R = A_{i_1}A_{i_2}\dots A_{i_R}$. Послідовність A_n , якій відповідає екстремальне значення критерію F , називається оптимальною. Якщо будь-який початковий відрізок оптимальної послідовності також оптимальний (в класі всіх послідовностей, складених з тих же елементів, і можливо, такий, що має ті ж початок і кінець, що і даний відрізок), то вважають, що для відповідної задачі справедливий принцип оптимальності.

Розглянемо зразок розв'язання задачі про комівояжера методом динамічного програмування:

1. Введення даних про пункти A_0, \dots, A_n і відстані між пунктами i та j d_{ij} ($d_{ij} = 0$ при $i = j$).
2. Обчислення всіх можливих варіантів відстаней, що складаються з трьох ділянок $A_0, A_{i_1}, A_{i_2}, A_{i_3}$. Вони групуються за останнім пунктом, з них залишаються ті варіанти, що об'єднують однакові пункти, але мають найменший шлях.

3. До тих варіантів, що залишились, додають ще четверту ділянку і повторюють процедуру з пункту 2. Це повторюється для п'ятої, шостої і т. д. ділянок, доки не повертаються в пункт A_0 . Той варіант (чи варіанти), що залишився, і визначає найкоротший шлях, по якому комівояжеру можна об'їздити всі міста A_i ($i=0, \dots, n$), якщо він почне та закінчить свою подорож в A_0 .

7.11. Варіаційні задачі

В варіаційних задачах місце змінних займають функції, а критерій оптимізації – деяка дійсна функція, що залежить від цих функцій і називається функціоналом.

В найпростішому випадку мають справу з функцією $y(x)$ однієї змінної і функціоналом вигляду

$$I = \int_{a_1}^{a_2} F(x, y, y') dx,$$

де a_1 і a_2 – константи, $F(x, y, z)$ – задана функція трьох дійсних змінних. При цьому припускається, що як сама функція $y = y(x)$, так і її похідна $y' = \frac{d}{dx} y(x)$ неперервні на відрізку $a_1 \leq x \leq a_2$.

В класичному варіаційному численні доводиться, що екстремум функції (мінімум або максимум) функціонала I досягається на функціях $y(x)$, які задовольняють диференціальне рівняння, тобто рівняння Ейлера:

$$F_y + F_{xy'} - F_{yy'} y' - F_{y'y'} y'' = 0.$$

Тут через F_y і F_{uv} ($u = x, y$ або y' , $v = y'$) позначені, відповідно, перша і друга частинні похідні функції $F(x, y, y')$ по змінній y і по змінним u, v .

7.12. Системна оптимізація

В практиці проектування великих систем і управління такими системами, як правило, використовується багато критеріїв. В ряді випадків їх вдається в той чи іншій спосіб звести до одного критерію і тим самим повернутися до вже дослідженого випадку однокритеріальної оптимізації.

Найпростіший спосіб такого зведення – так зване зважування критеріїв. Якщо $f_1(x), \dots, f_n(x)$ функції, що виражають значення

використовуваних критеріїв, то кожній з них, відповідно до відносної важливості критеріїв, вибирається додатний ваговий коефіцієнт λ_i . Операція зважування критеріїв (цільових функцій) $f_1(x), \dots, f_n(x)$ полягає в заміні їх єдиним критерієм (цільовою функцією) $f(x) = \lambda_1 f_1(x) + \dots + \lambda_n f_n(x)$.

Але для багатьох задач, що пов'язані з великими системами, подібне зведення виявляється практично неможливим, так що в процесі оптимізації доводиться мати справу з векторною (багатокритеріальною) цільовою функцією. При цьому припустима область M може змінюватись в процесі оптимізації. Більше того, в її цілеспрямованій зміні як раз і полягає основна змістовна сутність процесу оптимізації для подібного класу задач.

Наведемо одну з характерних формалізованих постановок задачі системної оптимізації.

Розв'язок відшукується безпосередньо в просторі K критеріїв оптимізації, які ми позначимо x_1 і x_2 . Процес розв'язання починається з того, що в заданому просторі K вибирається деяка точка A_0 з координатами a_0, b_0 – бажаний розв'язок задачі. За цим будуються початкові обмеження $F_1^{(0)}(x_1, x_2) \geq 0, \dots, F_n^{(0)}(x_1, x_2) \geq 0$, що задають початкову припустиму область P_0 . Прямою перевіркою встановлюється, чи належить точка A_0 області P_0 . В першому випадку в принципі може бути застосована звичайна (класична) процедура оптимізації або за одним з критеріїв x_1, x_2 , або за тією чи іншою їх комбінацією.

Але при системному підході застосовується звичайно запропонований Л. С. Понтрягіним спосіб, а саме: відповідно до моделі M вищого рівня, що управляє вибором критеріїв, точка A_0 виводиться з границь припустимої області P_0 , як це показано на рисунку 7.4.

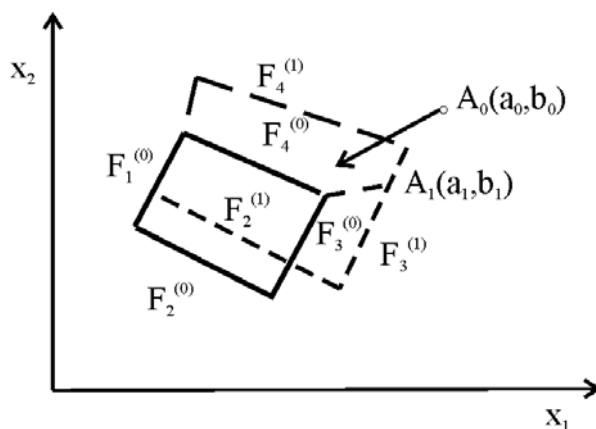


Рис. 7.4. Виведення точки A_0

Після цього виділяються ті обмеження, які не виконуються в точці A_0 (в випадку, що розглядається, ними будуть $F_3^{(0)}$ і $F_4^{(0)}$). Звертаючись до моделей M_3 і M_4 , які формують ці обмеження, в діалоговому режимі опробовуються ті чи інші рішення, що змінюють відповідні обмеження в потрібному напрямку (якщо така зміна можлива). Оптимальним при цьому вважається той напрямок, що зменшує абсолютну величину від'ємних відхилень $F_i^{(0)}(a_0, b_0)$ (в випадку, який розглядається, $F_3^{(0)}(a_0, b_0)$ і $F_4^{(0)}(a_0, b_0)$).

7.13. Застосування теорії графів до розв'язання оптимізаційних задач

Розв'язання багатьох технічних задач можливо методами теорії графів. Основні поняття та визначення теорії графів розглянуті в відповідних підручниках, які є в списку літератури в кінці цієї книжки. Тут ми обмежимося розглядом підходів до транспортної задачі та задачі комівояжера з застосуванням теорії графів.

Ці підходи базуються на відомій задачі з теорії графів про знаходження найкоротшого шляху між двома вершинами зв'язного неорієнтованого графу. До цієї задачі зводяться не тільки задача про комівояжера чи транспортна, а також, наприклад, багато задач оптимальної обробки деталей, оптимізації програмування, управління динамічними системами і т.д. В загальному вигляді задача формулюється так. Дано неорієнтований граф $G = (X, U)$, де X – множина вершин, U – множина ребер. Кожному ребру приписане деяке число $l(U) \geq 0$, що називається довжиною ребра (в транспортних задачах це може бути час чи вартість проїзду цим ребром). Треба для будь-яких вершин a і b графу G знайти такий шлях, що його довжина буде найкоротшою.

Загальне правило виявлення найкоротшого шляху в графі полягає в тому, щоб кожній вершині X приписати індекс λ_i , який дорівнює довжині найкоротшого шляху з даної вершини до кінцевої. Якщо, наприклад, спочатку взяти граф з ребрами одиничної довжини, то порядок дій буде такий:

1. Кінцевій вершині X_0 присвоюється індекс 0;
2. Усім вершинам, з котрих йде ребро до кінцевої вершини, присвоюється індекс 1;
3. Усім вершинам, що не мають індексів, та з котрих йде ребро в вершину з індексом λ_i , приписується індекс $\lambda_i + 1$. Цей процес продовжується до тих пір, поки не буде помічена початкова вершина.

Виявлені властивості дають можливість сформулювати такий спосіб знаходження найкоротшого шляху.

Нехай $X_n = a$ – початкова вершина з індексом λ_n . Шукаємо вершину X_{p_1} таку, що $\lambda_n - \lambda_{p_1} = l(X_{p_1}, X_n)$. Далі шукаємо вершину X_{p_2} таку, що $\lambda_{p_1} - \lambda_{p_2} = l(X_{p_2}, X_{p_1})$, і т.д. до тих пір, поки не дійдемо до кінцевої вершини. Шлях $\mu_0 = (X_n, X_{p_1}, \dots, X_{p_k}, X_0)$ – найкоротший. Приклад використання цього способу поданий на рисунку 7.6. Найкоротший шлях виділений подвійною рисою.

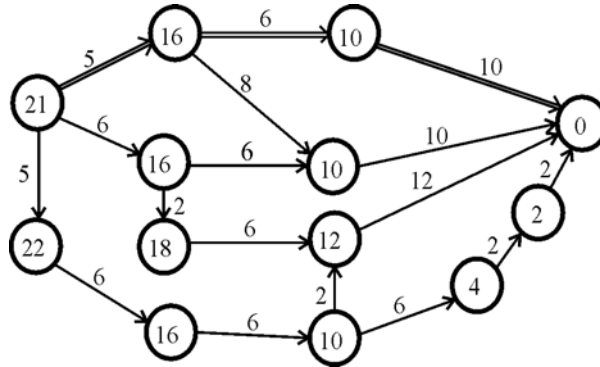


Рис. 7.6. Знаходження найкоротшого шляху в графі з ребрами довільної довжини

Очевидно, ці задачі про пошук найкоротшого шляху в графі аналогічні деяким задачам, що виникають в різних практичних областях. До неї зводяться задачі про побудову доріг, що з'єднують декілька міст найдешевшим чином, задачі про розбудову електромережі, нафтопроводів та ін. За допомогою графових підходів можна розв'язати транспортну задачу та задачу про комівояжера. Розглянемо задачу про комівояжера у вигляді графу. Вершини цього графу відповідають містам, а шляхи – дорогам, що ці міста з'єднують. Кожній дузі можна приписати довжину, що дорівнює відстані між містами. І далі дуже просто звести задачу комівояжера до задачі про пошук найкоротшого шляху. До задачі про комівояжера зводиться багато транспортних задач, задач оптимізації програмування, оптимізації порядку оброблення деталей та ін.

7.14. Застосування активних експериментів при ідентифікації моделей

Перевагою пасивних методів перед активними є те, що при їх використанні не потребується спеціального втручання в хід технологічного процесу. (Питання використання методів інтерполяції і апроксимації при обробці даних пасивних експериментів докладно розглянуті в попередній

главі.) Але ця перевага не компенсує недостатньої вірогідності математичних моделей, що отримуються. Тому в моделюванні розвинувся спеціальний напрямок – планування активного експерименту, що дозволяє ефективно використовувати наявні експериментальні ресурси.

Успішне проведення експерименту значною мірою залежить від правильного вибору плану експерименту, який також визначає статистичний аналіз результатів. Вибір методу аналізу залежить від алгебраїчної моделі, що придатна до різних методів обробки даних, і від відомого або припустимого розподілу імовірностей похибок вимірювань.

Як і в пасивному, в активному експерименті геометричним образом сукупності незалежних змінних \bar{X} і залежної змінної y є простір $(n + 1)$ -го виміру, де n – число незалежних змінних; $(n + 1)$ -й вимір стосується y . В цьому просторі залежності y від \bar{X} відповідає n -вимірною поверхню, яку називають поверхнею відгуку (результат дослідження розглядається як відгук системи на задану сукупність незалежних змінних або входів).

План експерименту вказує розташування дослідних точок в n -вимірному просторі незалежних змінних (або умови всіх дослідів, що їх треба провести). Найчастіше план експерименту задається в вигляді матриці планування – прямокутної таблиці, кожний рядок якої відповідає умовам певного дослідження, а n стовпців – значенням однієї з незалежних змінних в різних дослідженнях. В $(n + 1)$ -му стовпці наводяться одержані в дослідженнях значення залежної змінної. Приклад матриці планування експерименту:

№ дослідження	x_1	x_2	x_3	y
1	3	7	1	75
2	4	7	2	78
3	4	9	1	93
4	3	9	2	97

Тут в чотирьох дослідженнях досліджується вплив на вихід y трьох факторів-входів: x_1 , x_2 і x_3 .

При одній незалежній змінній план спрощений і найчастіше зводиться до рівномірного розподілу точок через рівні інтервали. В разі більш ніж трьох змінних ($n > 3$) план ускладнюється, і при його побудові прагнуть одержати деякі оптимальні властивості:

– максимальну точність (мінімальну дисперсію) в досліджуваній області при даній кількості дослідів або, при мінімальній кількості дослідів, задану точність;

– незалежність (некорельованість) оцінок впливу різних факторів.

Повний факторний експеримент

Найбільше розповсюджений такий порядок побудови планів: на першому етапі вибирається центр досліджуваної області (центр плану) і в нього переноситься початок координат; вибирається інтервал варіювання по кожній змінній – відстань по даній осі від центра до експериментальної точки. Вибір центра плану і інтервалу варіювання лежить поза математичною теорією. Цей етап повинен вирішуватись експериментатором на підставі знання об'єктів.

На наступному етапі здійснюють операцію приведення (кодування) змінних. Вона полягає в тому, що всі координати центра плану порівнюють з нулем, а інтервали варіювання Δ_j по кожній змінній приймають за одиницю. Кодовані змінні зручні тому, що вся обробка результатів дослідів проводиться в стандартній формі, яка не залежить від конкретних умов задачі; це істотно спрощує обчислення. Перехід від некованих (натуральних) значень змінних z_j до кодованих x_j і назад відбувається за формулою:

$$x_j = (z_j - z_{j0}) / \Delta_j$$

і відповідно

$$z_j = z_{j0} + x_j \Delta_j,$$

де z_{j0} – координата центра плану по j змінній. Розглянемо в наведених координатах найбільше поширений план 1-го порядку для двох незалежних змінних x_1 і x_2 . Розташування точок показано на рисунку 7.7. Точки розташовуються у вершинах квадрата, центр O якого збігається з центром плану, сторони паралельні осям і дорівнюють 2. Матриця планування має такий вигляд:

№ досліду	x_1	x_2
1	+1	+1
2	+1	-1
3	-1	+1
4	-1	-1

На практиці для скорочення запису часто замість "+1", "-1" пишуть "+", "-". Тоді матриця планування 1-го порядку буде мати вигляд:

№ досліду	x_1	x_2
1	+	+
2	+	-
3	-	+
4	-	-

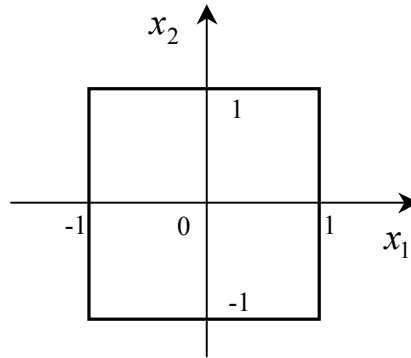


Рис. 7.7. План 1-го порядку для двох незалежних змінних

Основні особливості і властивості даного плану:

1. План побудований таким чином, що кожна незалежна змінна набуває в дослідах тільки два значення (+1 чи -1), тобто варіюється на двох рівнях (верхньому і нижньому). При цьому в чотирьох дослідах присутні всі можливі парні комбінації цих рівнів обох змінних. Чотири досліди, що поставлені за таким планом, називають повним факторним експериментом (ПФЕ) на двох рівнях для двох факторів; скорочено позначається ПФЕ 2^2 .

2. При проведенні дослідів за планом ПФЕ розрахунок коефіцієнтів рівняння методом найменших квадратів спрощується, причому з цих дослідів можна знайти коефіцієнти не тільки лінійного рівняння регресії, але й такого, що містить ще один член,

$$y = b_0 + b_1x_1 + b_2x_2 + b_{12}x_1x_2.$$

В теорії експерименту члени, що містять добутки факторів, називають взаємодіями. Такі члени необхідно враховувати, оскільки вони показують, наскільки вплив одного фактора залежить від значення іншого.

3. Оптимальність використання експериментальних ресурсів: при заданих точності експерименту, кількості дослідів і межах вимірів факторів рівняння регресії виявляється більш точним, ніж те, що одержується з експерименту з іншим розташуванням точок, що досліджуються.

4. Важлива особливість плану – некорельованість факторів. Вона базується на ортогональності матриці планування. Всі стовпці матриці планування (і весь план в цілому) ортогональні. З ортогональності однозначно випливає взаємна незалежність (некорельованість) всіх факторів і, відповідно, всіх коефіцієнтів рівняння регресії.

Вибір матриці планування у вигляді повного факторного експерименту при використанні в якості моделі полінома першого степеня забезпечує оптимальне планування на підставі таких властивостей: всі

розрахунки і обчислення проводяться просто; коефіцієнти регресії визначаються незалежно один від одного; дисперсії всіх коефіцієнтів регресії рівні і мінімальні; дисперсія вихідного параметра не залежить від обертання системи координат в центрі плану, а тільки від радіуса досліджуваної сфери факторного простору (властивість рототабельності).

Правильний вибір центра експерименту, інтервалів і рівнів варіювання факторів має вирішальне значення для достовірності побудованої математичної моделі. Основна вимога до інтервалу варіювання полягає в тому, щоб він не був меншим за подвійну квадратичну похибку фактора. Ця вимога пов'язана з тим, що інтервал між двома сусідніми рівнями повинен значно впливати на вихідний параметр. Звичайно інтервал варіювання вибирається на підставі апріорної інформації і потім уточнюється після одержання математичної моделі. Вдалий вибір інтервалу варіювання факторів гарантує достовірність математичної моделі об'єкта. Якщо інтервал варіювання вибраний невдало і модель неадекватна, то для уточнення моделі необхідно повторити експерименти.

Важливим елементом розробки плану експерименту є вибір числа рівнів для кожного фактора. Найбільше розповсюдження одержало планування факторів на двох рівнях, коли в якості рівнів використовуються верхня і нижня межі інтервалу варіювання. Постановка експериментів за такими планами називається дворівневим повним факторним експериментом типу 2^n , де n – число факторів. Тоді для двох факторів число експериментів $2^2 = 4$, для трьох $2^3 = 8$ і т.д.

Розглянемо побудову матриці планування для повного факторного експерименту при трьох факторах (ПФЕ, 2^3). Щоб перебрати всі комбінації для трьох факторів на двох рівнях, випишемо два рази комбінації для двох факторів (ПФЕ, 2^2), один раз в поєднанні зі значенням $x_3 = +1$, другий – зі значенням $x_3 = -1$ (рисунок 7.8). Виходить такий план (матриця ПФЕ, 2^3):

№ досліду	1	2	3	4	5	6	7	8
x_1	+1	+1	-1	-1	+1	+1	-1	-1
x_2	+1	-1	+1	-1	+1	-1	+1	-1
x_3	+1	+1	+1	+1	-1	-1	-1	-1

Розглянемо приклад розробки математичної моделі за результатами активного експерименту (цей приклад взято з книги Семесенко М.П., Пелых С.Т. ЭВМ в управлении литейными процессами. –Донецк: Донбасс, 1975 – 97 с.). Розглядається експеримент з трьома вихідними y_1, y_2, y_3 та трьома вхідними x_1, x_2, x_3 змінними. Вихідні дані для проведення серії експериментів:

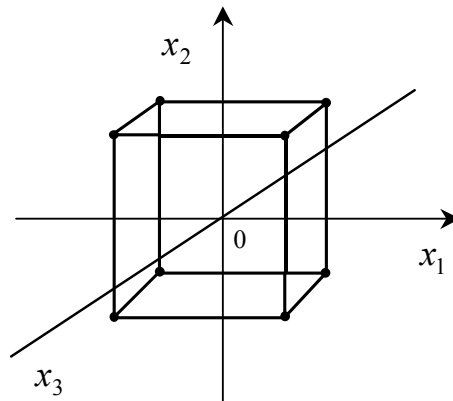


Рис. 7.8. Матриця планування для повного факторного експерименту

Параметр		x_1	x_2	x_3
рівень:	Середній	15	50	75
	Верхній (1)	20	60	100
	Нижній (-1)	10	40	50
Інтервал		5	10	25

Результати реалізації активного ПФЕ:

№ досліду	Вхідні параметри			Вихідні параметри		
	x_1	x_2	x_3	y_1	y_2	y_3
1	1	1	1	1.5	92	6.4
2	-1	1	1	0.5	163	5.5
3	1	-1	1	1	110	4.8
4	-1	-1	1	0.7	142	3.4
5	1	1	-1	1.2	77	6.7
6	-1	1	-1	0.3	152	5.8
7	1	-1	-1	0.4	105	5.1
8	-1	-1	-1	0.3	143	4.0

Після проведення розрахунків за формулами

$$b_0 = \frac{1}{N} \sum_{j=1}^N y_j x_{j0},$$

$$b_i = \frac{1}{N} \sum_{j=1}^N y_j x_{ji},$$

$$b_{ik} = \frac{1}{N} \sum_{j=1}^N y_j x_{ji} x_{jk}$$

отримані такі коефіцієнти регресії:

	b_0	b_1	b_2	b_3	b_{12}	b_{13}	b_{23}
y_1	0,73	0,28	0,0145	0,019	0,017	0,004	-0,0004
y_2	123	-27	-0,193	0,375	-0,9	0,125	0,0250
y_3	5,19	0,54	0,085	-0,019	-0,008	0,003	0,0003

Перевірка адекватності моделі: припустимо, що при проведенні паралельних експериментів по кожній комбінації параметрів при середньоквадратичних відхиленнях $\sigma_{y_1} = 0,086$, $\sigma_{y_2} = 4,5$, $\sigma_{y_3} = 0,2$ середньоквадратичні відхилення коефіцієнтів регресії становлять, відповідно, $\sigma_{b_1}^{(y_1)} = 0,003$; $\sigma_{b_2}^{(y_2)} = 0,16$; $\sigma_{b_3}^{(y_3)} = 0,006$. Тоді $\Delta b^{(y_i)} = t_{\alpha, N} * \sigma_b^{(y_i)}$, де $t_{\alpha, N}$ - критерій Стюдента, що визначається за статистичними таблицями і залежить від рівня значущості α та кількості експериментів N . Тоді для $t_{0,05;8} = 2,3$ можна отримати інтервали довіри відповідно для коефіцієнтів рівняння

$$\Delta b^{(y_1)} = 0,0069; \quad \Delta b^{(y_2)} = 0,368; \quad \Delta b^{(y_3)} = 0,014.$$

Якщо виключити для кожного з рівнянь коефіцієнти, що менше за ці інтервали, отримаємо остаточні рівняння:

$$y_1 = 0,73 + 0,28x_1 + 0,0145x_2 + 0,019x_3 + 0,019x_1x_2;$$

$$y_2 = 123 - 27x_1 + 0,375x_3 - 0,9x_1x_2;$$

$$y_3 = 5,19 + 0,54x_1 + 0,085x_2 - 0,019x_3.$$

Табличне значення критерію Фішера $F_{\alpha; m; N} = F_{0,05; 3; 8} = 4,1$. Порівняння цього значення із значенням, що розраховано, показує, що $F_{розр} < F_{табл}$. Можна зробити висновок, що ці моделі статистично адекватні.

Дробовий факторний експеримент (ДФЕ)

Дворівневий план ефективний (особливо при $n > 3$), коли для перевірки адекватності моделі мають достатню кількість ступенів вільності, тобто є перевищення числа дослідів N над числом коефіцієнтів моделі $(n + 1)$, що визначаються. Число 2^n визначає об'єм вибірки при повному факторному експерименті. При певній постановці задачі іноді параметри доводиться варіювати не на двох, а на більшій кількості рівнів. Якщо число потрібних рівнів позначити через K , то об'єм повного факторного експерименту визначиться виразом K^n . Великий об'єм вибірки дає можливість точно визначити коефіцієнти моделі і мати достатньо ступенів вільності для перевірки адекватності моделі. Але це призводить до великих експериментальних і обчислювальних робіт. Наприклад, якщо в чотирифакторному експерименті фактори A і B мають по два рівні, фактор C – три і фактор D – чотири рівні, то число точок плану дорівнює $2 * 2 * 3 * 4 = 48$.

При збільшенні числа факторів і числа рівнів кожного фактора істотно зростає число точок плану, що створюють повну факторну решітку. Наприклад, якщо б кожний з семи факторів в деякому модельному прикладі мав тільки два рівні, то для побудови повного факторного плану було б потрібно $2^7 = 128$ точок. Таким чином, повний факторний план приводить до надмірних витрат часу і засобів.

В разі неповного дослідження (наприклад, дослідження лише головних ефектів і двофакторних взаємодій без розгляду взаємодій більш високого порядку) використовують неповні факторні плани, які дозволяють розв'язати поставлену задачу і потребують меншої кількості дослідів порівняно з повним факторним планом.

В будь-якому плані з числом дослідів, меншим, ніж у повного факторного плану, присутні ефекти змішування.

Звичайно в ході експерименту цікавими є головні ефекти, і важливо, щоб вони не змішувались між собою. Практично в усіх неповних факторних планах, що застосовуються, головні ефекти змішуються лише з ефектами взаємодії високого порядку. Таким чином, при оцінці головних ефектів за допомогою цих планів припускається (в крайньому разі, попередньо), що взаємодії, які можуть змішуватися з головними ефектами, дорівнюють нулю або достатньо малі.

Кількість експериментів можна скоротити, якщо заздалегідь домовитись про структуру моделі. Припустимо, для задачі, що досліджується, достатньо мати лінійну модель об'єкта. Тоді в разі наявності трьох змінних повний факторний експеримент дозволяє знайти вільний член, три коефіцієнти при лінійних членах і чотири ступені вільності для перевірки адекватності. При $m = 6$ загальний об'єм експериментів дорівнює 64, з них сім експериментів використовується для визначення вільного члена і коефіцієнтів при лінійних членах, а 57 – для

перевірки адекватності моделі. В цьому випадку, як ми бачимо, є надлишок інформації для знаходження адекватності моделі процесу. При цьому доцільно ставити не всі 2^m експериментів, а менше, в залежності від вибраного числа ступенів вільності визначення адекватності моделі і від вибраної структури моделі. Зменшити число експериментів можна, якщо прирівняти рівні однієї або декількох змінних добутку рівнів інших змінних, коли їх мінімальні і максимальні рівні прийняті, відповідно, за -1 і $+1$.

Нехай є чотири змінних. Повний факторний експеримент для такої кількості змінних дорівнює 16. Для лінійної структури моделі достатньо п'яти експериментів. При цьому рівень четвертої змінної можна брати за знаком добутку перших трьох змінних, тобто $x_4 = x_1 * x_2 * x_3$, де x_1, x_2, x_3 можуть приймати значення -1 чи $+1$. Якщо перший експеримент ставиться таким чином, що перша змінна знаходиться на нижньому рівні, тобто дорівнює -1 , друга також на нижньому рівні, а третя – на верхньому, то четвертий параметр ставиться на верхній рівень. Число експериментів в такому випадку скорочується до восьми. Вісім експериментів достатньо для визначення вільних членів, коефіцієнтів при лінійних членах і для оцінки адекватності моделі.

Такий план називається дробовою реплікою (ДР) від повного факторного експерименту. В залежності від постановки задачі повний факторний експеримент можна скорочувати до дробових реплік різного розміру. При наявності семи параметрів повний факторний експеримент 2^7 , наприклад, можна скоротити до дробової репліки розміру 2^3 , якщо прийняти, що $x_4 = x_1 * x_2$; $x_5 = x_1 * x_3$; $x_6 = x_2 * x_3$; $x_7 = x_1 * x_2 * x_3$.

В загальному випадку при наявності m параметрів і скороченні повного факторного експерименту на k ступенів вийде дробовий факторний експеримент об'ємом 2^{m-k} .

Для побудови дробових реплік використовують спеціальні алгебраїчні вирази, які полегшують виявлення змішаних ефектів. Їх називають генерувальними співвідношеннями чи визначальними контрастами. Генерувальним називається співвідношення, що показує, які взаємодіючі фактори замінені новими (воно генерує, або створює ДР). Так, вищерозглянуте планування типу 2^{3-1} (піврепліка від 2^3), що задавалось одним з таких генерувальних співвідношень: $x_3 = x_1 * x_2$ або $x_3 = -x_1 * x_2$.

З генерувальними співвідношеннями можна проводити алгебраїчні операції: множити обидві частини рівняння на будь-які ефекти – лінійні, подвійні, потрійні та інші взаємодії. При цьому фактор, що піднесений до квадрата або до іншого парного ступеня, замінюють одиницею ($x_j^{2v} = 1$; $v = 1, 2, 3, \dots$). Помноживши генерувальні співвідношення для плану

2^{3-1} на x_3 , одержуємо $x_3^2 = x_1 * x_2 * x_3$; $x_3^2 = -x_1 * x_2 * x_3$, або, якщо врахувати на те, що сказано вище, $x_1 * x_2 * x_3 = 1$; $-x_1 * x_2 * x_3 = 1$.

Такі добутки називають визначальним контрастом. Визначальний контраст дає можливість встановити розв'язувальну здатність дробової репліки, тобто знайти, які з коефіцієнтів є незмішаними оцінками факторів. За прийнятим визначальним контрастом одержують співвідношення, які задають змішані оцінки для даної дробової репліки. Для цього кожний фактор множать на визначальний контраст.

Коефіцієнти рівняння регресії для ПФЕ і ДР визначаються за однаковими формулами. При використанні вимірних змінних x_1 переходять до безвимірних (нормованих) змінних:

$$\bar{x}_i = (x_i - x_i^*) / \Delta x_i.$$

Тут \bar{x}_i визначені через змінну на вихідному (основному) рівні +1. Для повних факторних планів (ПФП) і їх ДР

$$\sum_{u=1}^n \bar{x}_{iu} \bar{x}_{ju} = 0, \quad j < > i;$$

$$\sum_{u=1}^n \bar{x}_{0u} \bar{x}_{iu} = \sum_{u=1}^n \bar{x}_{iu} = 0;$$

$$\sum_{u=1}^n (\bar{x}_{iu})^2 = n.$$

При використанні нормованих змінних спрощується розрахунок коефіцієнтів b і їх дисперсії. Для ПФП і ДР знаходимо

$$b_0 = \frac{y_0}{n}; \quad b_i = \frac{y_i}{n}; \quad b_{ij} = \frac{y_{ij}}{n};$$

$$\sigma_{bi}^2 \approx s_{bi}^2 = \frac{s^2}{n} = \frac{\sum_{u=1}^n (y_{u\bar{e}} - y_{u\bar{p}})^2}{v \cdot n},$$

де надійний інтервал $b_i = s_{bi} * t(p)$; $v = n - (p + 1)$.

Наведені співвідношення характеризують нормалізоване рівняння регресії $y = b_0 + b_1 \bar{x}_1 + \dots + b_p \bar{x}_p$.

Активна ідентифікація нелінійних моделей

Якщо математична модель процесу нелінійна, то, як і при пасивних методах, вона апроксимується поліномом. Однак при одержанні математичної моделі кількість необхідних дослідів при зростанні числа членів цього полінома швидко зростає. В зв'язку з цим потрібно інакше розв'язувати питання про число рівнів, центр плану експерименту і принципи оптимальності планів, що застосовуються. Розв'язок цих питань здійснюється різними методами. Найчастіше в інженерній практиці для опису нелінійної області використовуються методи центрального композиційного рототабельного і ортогонального планування. При плануванні за допомогою цих методів необхідно перш за все вибрати нульову точку, число рівнів і принцип оптимальності.

За нульову точку приймається центр плану, що вже був використаний раніше (наприклад, ПФЕ), який дає неадекватну лінійну модель (значущими виявляються ефекти взаємодії і квадратичні). При цьому проведені експерименти доповнюються спеціальними дослідями (такі плани називаються центральними композиційними). При використанні математичної моделі в вигляді полінома другого порядку двома рівнями варіювання факторів обмежуватись не можна. Але плани на трьох рівнях неекономічні з точки зору кількості дослідів. Якщо доповнити дворівневий план ПФЕ визначеними точками факторного простору, можна одержати оптимальний план. Ядро центрального композиційного плану складає ПФЕ типу 2^n при $n < 5$. Якщо $n > 5$, то користуються дробовими репліками, які забезпечують роздільне визначення лінійних ефектів і ефектів взаємодії. План ПФЕ доповнюють деякою кількістю зіркових точок, координати яких залежать від прийнятого принципу оптимальності. Загальна кількість дослідів при такому плануванні визначається формулою $N = 2^n + 2n + n_0$, де доданки – відповідно, число дослідів ПФЕ, зіркових і нульових точок.

Розглянемо побудову найбільш розповсюдженого типу планів другого порядку – центрального композиційного плану: центральні – внаслідок симетричності відносно центра плану, композиційні – оскільки вони komponуються додаванням визначеного числа дослідів до плану 1-го порядку. В цьому велика перевага таких планів: якщо рівняння 1-го порядку неадекватно описує об'єкт, то не треба ставити всі досліди заново, а достатньо додати точки – добудувати план до плану 2-го порядку, як показано на рисунок 7.9 ($n = 2$).

До точок повного факторного експерименту додаються точки в центрі плану (одна або декілька паралельних) і точки, що розташовані на всіх осях координат на відстанях $\pm \alpha$ від центра. Точки на осях називають зірковими точками, а величину α – зірковим плечем. Для різних планів зіркове плече може бути різним.

Матриця такого плану для двох незалежних змінних (план 2-го порядку при $n = 2$) має вигляд:

x_1	+1	+1	-1	-1	0^*	0^*	$+\alpha^*$	$-\alpha^*$	0^{**}
x_2	+1	-1	+1	-1	$+\alpha^*$	$-\alpha^*$	0^*	0^*	0^{**}

* – зіркова точка; ** – центральна точка.

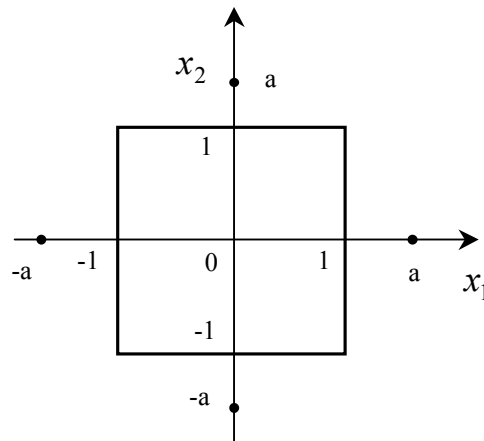


Рис. 7.9. Центральний композиційний план

Загальне число точок плану 2-го порядку дорівнює $2^n + 2n + m = 9$, де $2n$ – число зіркових точок, m – число точок в центрі. Якщо план 1-го порядку – дробовий, то число точок плану 2-го порядку дорівнює $2^{n-k} + 2n + m$.

При рототабельному плануванні і повному факторному експерименті зіркове плече $\alpha = \frac{n}{2^4}$; для дробового факторного експерименту $\alpha = 2^{\frac{n-p}{4}}$, де p – дробовість репліки ($p = 1$ – піврепліка, $p = 2$ – чверть репліки і т.д.).

Для вибору зіркового плеча, числа зіркових і нульових точок користуються даними таблиці:

Параметри плану	Число змінних n				
	для ПФЕ типу 2^n				Для ДФЕ типу 2^{n-1}
	2	3	4	5	5
Число дослідів ядра матриці	4	8	16	32	16
Число зіркових точок	4	6	8	10	16
Число нульових точок	5	6	6	10	6
Зіркове плече	1,414	1,682	2,000	2,378	2,000

При рототабельному плануванні завдяки експериментальним точкам в центрі плану дисперсія передбачуваного значення всередині області експериментування постійна і не залежить від відстані до центра плану.

При обробці планів 2-го порядку потребується значна кількість комп'ютерних обчислень.

При ортогональному плануванні до факторного експерименту або дробової репліки додають $2p+1$ дослідів (p – число змінних), причому один з них – "центральний" ($\bar{x}_1 = \bar{x}_2 = \dots = \bar{x}_p = 0$), а в $2p$ – "зіркові". В "зіркових" дослідах кожна з нормованих змінних по черзі приймає значення $\pm\alpha$, а для решти змінних заданий основний рівень ($\bar{x}_1 = 0; j < > i$). При кількості змінних $n = 2; 3$ і 4 значення α_0 відповідно дорівнюють 1; 1,215 і 1,414.

За результатами ортогонального планування визначають коефіцієнти рівняння регресії:

$$b_i = \frac{\sum_{u=1}^n \bar{x}_{iu} y_u}{\sum_{u=1}^n (\bar{x}_{iu})^2}; \quad b_{ij} = \frac{\sum_{u=1}^n (\bar{x}_{iu} \bar{x}_{ju}) y_u}{\sum_{u=1}^n (\bar{x}_{iu} \bar{x}_{ju})^2}; \quad b_{ii} = \frac{\sum_{u=1}^n (\bar{x}_{iu})^2 y_u}{\sum_{u=1}^n (\bar{x}_{iu})^2};$$

$$b_0 = \frac{\sum_{u=1}^n y_u}{n} - b_{11} \frac{\sum_{u=1}^n (\bar{x}_{1u})^2}{n} - \dots - b_{pp} \frac{\sum_{u=1}^n (\bar{x}_{pu})^2}{n}.$$

В ході оцінки адекватності при рототабельному і ортогональному плануванні розрахункове значення критерію Фішера визначають як відношення дисперсії адекватності до дисперсії дослідів:

$$F = \frac{s_{ad} / f_{ad}}{s_0 / f_0} = \frac{s_{ad}^2}{s_0^2}$$

і порівнюють з табличним F_T для ступенів вільності f_{ad} і f_0 .

Контрольні запитання та завдання

1. Сформулюйте класичну постановку задачі оптимізації.
2. В чому полягає задача оптимального проектування?
3. За якими ознаками класифікуються задачі оптимізації?
4. Яка задача дістала назву математичного програмування?

5. Які особливості мають задачі нелінійного, стохастичного, динамічного, квадратичного, сепарабельного, геометричного програмування?
6. Чим відрізняються задачі багатопараметричної і багатокритеріальної оптимізації?
7. Сформулюйте умови стаціонарності для задачі гладкої оптимізації.
8. Запишіть та поясніть умови Куна-Такера.
9. Дайте характеристики та порівняйте чисельні методи розв'язання задачі гладкої оптимізації.
10. Спробуйте розв'язати задачу оптимізації іншими методами, окрім методу градієнтного спуску.
11. Як звести загальну задачу оптимізації до задачі без обмежень?
12. Які існують методи опуклої оптимізації?
13. Зробіть постановку задачі лінійного програмування. В яких випадках вона не має розв'язку?
14. Дайте характеристику методу розгалужень і меж розв'язання задачі лінійної оптимізації.
15. В яких випадках і як застосовується теорія ігор для розв'язання задач дискретної оптимізації? Як вибирається стратегія гравця?
16. Сформулюйте принципи оптимізації Беллмана.
17. Яку задачу називають задачею комівояжера?
18. Складіть програми пошуку найкоротшого шляху в графах з ребрами одиничної та довільної довжини.
19. Чим відрізняються активні та пасивні експерименти?
20. Що таке план експерименту? Як він задається? Чим відрізняються дробовий та повний факторні експерименти?
21. Як розраховуються коефіцієнти рівняння регресії?
22. Як визначити необхідну кількість випробувань при проведенні активних експериментів?

Список літератури

1. Абрамовиц М., Стиган И. Справочник по специальным функциям. – М.: Наука, 1979.
2. Батищев Д. И. Методы оптимального проектирования. – М.: Радио и связь, 1984.
3. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. – М.: Наука, 1987.
4. Воеводин В. В. Матрицы и вычисления. – М.: Наука, 1984 .
5. Глушков В. М. Основы безбумажной информатики. – М.: Наука, 1987.
6. Демидович Б. П., Марон И. А. Основы вычислительной математики. – М. Наука, 1970.
7. Дубовий В. М., Кветний Р. Н. Основи застосування ЕОМ у інженерній діяльності. – К.: ІСДО України, 1994.
8. Дубовой В. М., Кветний Р. Н. Програмування комп'ютеризованих систем управління та автоматики. – Вінниця: ВДТУ, 1997.
9. Дубовой В. М., Кветний Р. Н. Програмування персональних комп'ютерів систем управління. – Вінниця.: ВДТУ, 1999.
10. Канторович Л. В., Горстков А. Б. Оптимальные решения в экономике. – М.: Наука, 1972.
11. Кветний Р. Н. Математическое моделирование в задачах проектирования средств автоматики и информационно-измерительной техники. – К.:УМК ВО,1989.
12. Кветний Р. Н., Маликов В. Т. Информационная теория измерений: от модели к изделию. – М.: Знание, 1988.
13. Коллатц Л. Функциональный анализ и вычислительная математика. - М.: Мир, 1969.
14. Краскевич В. Е., Зеленский К. Х., Гречко В. И. Численные методы в инженерных исследованиях. – К.: Вища школа, 1986.
15. Мак-Кракен Д., Дорн У. Численные методы и программирование на ФОРТРАНЕ. – М.: Мир, 1977.
16. Маликов В. Т., Кветний Р. Н. Вычислительные методы и применение ЭВМ. – К.: Вища школа, 1989.
17. Марчук Г. И., Агошков В.И. Введение в проекционно–сеточные методы. – М.: Наука, 1982.
18. Ракитский Ю. В., Устинов С. М., Черноруцкий И. Г. Численные методы решения жестких систем. – М.: Наука, 1979.
19. Саати Т. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. – М.: Мир, 1973.
20. Самарский А. А. Введение в численные методы. – М.: Наука, 1987.
21. Самарский А. А. Теория разностных схем. – М.: Наука, 1977.
22. Скурихин В. Н., Шифрин В. Б., Дубровский В. В. Математическое моделирование. – К.: Техніка, 1983.

23. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений. – М.: Мир, 1980.
24. Шуп Т. Решение инженерных задач на ЭВМ. – М.: Мир, 1982.
25. Чабан В. Чисельні методи. – Львів: В-во Національного ун-ту «Львівська політехніка», 2001.



Роман Кветний

Народився в 1955 році.

Доктор технічних наук, професор, Академік Української Технологічної Академії, Старший член Всесвітнього інституту інженерів з електроніки та електротехніки (IEEE), завідувач кафедри автоматичної та інформаційно-вимірювальної техніки Вінницького державного технічного університету.

Спеціаліст в області математичного моделювання, автор біля 200 наукових праць (статей, доповідей та винаходів), 12 монографій та навчальних посібників; має низку літературних творів (віршів, прози, публіцистики), що видані окремою книгою.

Наукові праці та дослідження присвячені розробці математичних методів моделювання складних систем в умовах невизначеності: ймовірнісні методи (докторська дисертація в 1990 році), інтервальне моделювання, фрактальна інтерполяція, методи оцінки ризику та ін., ранні роботи: приладобудування та вимірювальна техніка (зокрема, ультразвуковий метод).

Навчальне видання

Кветний Р. Н.

МЕТОДИ КОМП'ЮТЕРНИХ ОБЧИСЛЕНЬ

Навчальний посібник

Оригінал-макет підготовлено автором

Редактор В. О. Дружиніна

Підписано до друку

Формат 29,7x42 1/4 Гарнітура Times New Roman

Друк різнографічний Ум. друк. арк.

Тираж 200 прим.

Зам. №

Віддруковано в комп'ютерному інформаційно-видавничому центрі
Вінницького державного технічного університету
21021, м. Вінниця, Хмельницьке шосе, 95, ВДТУ, ГНК, 9-й поверх
Тел. **(0432) 44-01-59**